



# Red Hat 3scale API Management 2.11

## Admin Portal Guide

Manage aspects related to Red Hat 3scale API Management.



## Red Hat 3scale API Management 2.11 Admin Portal Guide

---

Manage aspects related to Red Hat 3scale API Management.

## Legal Notice

Copyright © 2023 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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

The OpenStack<sup>®</sup> 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 provides the information to manage Red Hat 3scale API Management.

## Table of Contents

<b>PREFACE</b> .....	<b>7</b>
<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>8</b>
<b>PART I. ACCOUNT SETTINGS</b> .....	<b>9</b>
<b>CHAPTER 1. ACCOUNT CONFIGURATION</b> .....	<b>10</b>
1.1. ADD YOUR COMPANY INFORMATION	10
1.2. SELECT YOUR PREFERRED TIME ZONE	10
<b>CHAPTER 2. RED HAT SINGLE SIGN ON FOR THE 3SCALE ADMIN PORTAL</b> .....	<b>11</b>
2.1. ENABLE RH-SSO OR AUTH0 MEMBER AUTHENTICATION	11
2.1.1. RH SSO prerequisites	11
2.1.2. Auth0 prerequisites	11
2.1.3. Enable RH-SSO	11
2.2. USING RH-SSO WITH 3SCALE	12
2.3. REDIRECTING A 3SCALE LOGIN TO A RH-SSO OPTION	12
2.3.1. Prerequisites	12
2.3.2. Required steps	12
<b>CHAPTER 3. INVITING USERS AND MANAGING RIGHTS</b> .....	<b>14</b>
3.1. NAVIGATE TO USER ADMINISTRATION	14
3.2. SEND AN INVITATION	14
3.3. ACCEPT THE INVITATION	14
3.4. GIVE NEW USERS RIGHTS	14
<b>CHAPTER 4. NOTIFICATIONS</b> .....	<b>16</b>
4.1. TYPES OF NOTIFICATIONS	16
4.2. VISIBILITY	16
4.3. SUBSCRIBING TO NOTIFICATIONS BY EMAIL	16
4.4. WEB NOTIFICATIONS	16
<b>CHAPTER 5. PERSONAL SETTINGS</b> .....	<b>17</b>
<b>CHAPTER 6. TOKENS</b> .....	<b>18</b>
6.1. ACCESS TOKENS	18
6.2. CREATING ACCESS TOKENS	18
6.3. USING ACCESS TOKENS	18
6.3.1. Service tokens	19
<b>PART II. MANAGING DEVELOPER ACCOUNTS</b> .....	<b>20</b>
<b>CHAPTER 7. ADDING DEVELOPERS</b> .....	<b>21</b>
7.1. CREATE A NEW DEVELOPER ACCOUNT	21
7.2. SET UP APPLICATIONS	21
7.3. NOTIFY THE DEVELOPER	21
<b>CHAPTER 8. APPROVING DEVELOPERS</b> .....	<b>22</b>
8.1. APPROVE FROM EMAIL NOTIFICATION	22
8.2. ACCOUNT APPROVAL	22
8.3. SERVICE APPROVAL	22
8.4. APPLICATION APPROVAL	23
<b>CHAPTER 9. CHANGING PLANS FOR AN APP</b> .....	<b>25</b>
9.1. CHANGE ACCOUNT PLANS	25

---

9.2. CHANGE SERVICE PLANS	25
9.3. CHANGE APPLICATION PLANS	26
9.3.1. More information	27
<b>CHAPTER 10. CONTACTING DEVELOPERS</b>	<b>28</b>
10.1. LOCATE THE RELEVANT APPLICATION AND ACCOUNT IN THE SYSTEM	28
10.2. SEND INTERNAL MESSAGES TO DEVELOPERS	28
10.3. CONTACT BY OTHER MEANS	29
<b>CHAPTER 11. CUSTOMIZE PLANS</b>	<b>30</b>
11.1. CHOOSE THE ACCOUNT	30
11.2. SELECT THE APPLICATION	30
11.3. CUSTOMIZE THE APPLICATION PLAN	30
11.3.1. More information	31
<b>CHAPTER 12. ENABLE SIGNUP</b>	<b>32</b>
<b>CHAPTER 13. FINDING APPLICATIONS</b>	<b>33</b>
13.1. GET THE INFORMATION YOU NEED	33
13.2. SEARCH FOR THE APPLICATION	33
13.3. ACCESS APPLICATION INFORMATION	33
<b>CHAPTER 14. INVITING DEVELOPERS</b>	<b>35</b>
<b>CHAPTER 15. UNSUBSCRIBING DEVELOPERS FROM A SERVICE</b>	<b>36</b>
15.1. UNSUBSCRIBING A SINGLE DEVELOPER FROM SERVICES	36
15.2. UNSUBSCRIBING MULTIPLE DEVELOPERS FROM SERVICES	36
<b>CHAPTER 16. SUSPENDING APPLICATIONS</b>	<b>37</b>
16.1. FIND THE APPLICATION	37
16.2. DISABLE THE APPLICATION	37
16.3. CONTACT THE DEVELOPER	37
<b>CHAPTER 17. DELETING APPLICATIONS</b>	<b>39</b>
<b>CHAPTER 18. DELETING AN API</b>	<b>40</b>
<b>PART III. ACCESS CONTROL</b>	<b>41</b>
<b>CHAPTER 19. DESIGNATING METHODS AND ADDING METRICS FOR CAPTURING USAGE DETAILS</b>	<b>42</b>
19.1. ADDING METHODS TO PRODUCTS AND BACKENDS	42
19.2. ADDING METRICS TO PRODUCTS AND BACKENDS	43
19.3. ALTERNATIVES FOR IMPORTING METHODS AND METRICS	44
19.4. ADDING MAPPING RULES TO METHODS AND METRICS	44
19.5. ADDITIONAL RESOURCES	45
<b>CHAPTER 20. APPLICATION PLANS</b>	<b>46</b>
20.1. HOW TO CREATE AN APPLICATION PLAN	46
20.2. SETTING UP A DEFAULT APPLICATION PLAN	46
<b>CHAPTER 21. PROVISIONING PAID PLANS</b>	<b>47</b>
21.1. DECIDING YOUR PRICING MODEL	47
21.2. CONFIGURING AN APPLICATION PLAN WITH YOUR PRICING RULES	47
21.3. CREATING FURTHER PRICING TIERS	48
21.4. PROVISIONING THE PAID PLANS	48
21.5. ADDITIONAL REFERENCES	48

<b>CHAPTER 22. PROVISIONING RATE LIMITS</b> .....	<b>49</b>
22.1. CONFIGURING THE APPLICATION PLAN	49
22.2. SETTING THE RATE LIMITS	49
22.3. PUTTING THE NEW RATE LIMITS INTO ACTION	49
22.4. MORE INFORMATION	49
<b>PART IV. BILLING</b> .....	<b>50</b>
<b>CHAPTER 23. CONFIGURING BILLING SETTINGS</b> .....	<b>51</b>
23.1. BILLING MODES (CHARGING & GATEWAY)	51
23.2. CHARGING ENABLED (CHARGING & GATEWAY)	51
23.3. CURRENCY (CHARGING & GATEWAY)	51
23.4. INVOICE FOOTNOTE (CHARGING & GATEWAY)	51
23.5. TEXT TO SHOW IF VAT/SALES TAX IS 0% (CHARGING & GATEWAY)	51
23.6. YAML CONFIGURATION FOR CURRENCIES	51
23.6.1. Changing the currencies configuration in OpenShift	52
23.6.2. Verifying the new currencies	53
23.7. BILLING PERIODS FOR INVOICE IDS (CHARGING & GATEWAY)	53
23.8. CREDIT CARD POLICIES	54
<b>CHAPTER 24. CREDIT CARD GATEWAYS FOR PAYMENTS</b> .....	<b>55</b>
24.1. CREDIT CARD GATEWAYS SUPPORTED BY 3SCALE	55
24.2. CONFIGURING STRIPE AS A CREDIT CARD GATEWAY	55
24.2.1. Generating an access token with the Billing API scope in the 3scale Admin Portal	56
24.2.2. Obtaining the keys and the webhook secret from Stripe	56
24.2.3. Configuring charging in the 3scale Admin Portal	57
24.2.4. Editing credit card details in the 3scale Developer Portal	57
24.2.5. Updating the text of unsuccessfully charged email response	57
24.3. CONFIGURING BRAINTREE AS A CREDIT CARD GATEWAY	58
24.4. ALLOWING PAYMENTS OF REJECTED INVOICES VIA THE DEVELOPER PORTAL	60
24.5. TROUBLESHOOTING ISSUES WITH CREDIT CARD GATEWAYS	60
<b>CHAPTER 25. PRICING</b> .....	<b>62</b>
25.1. PRICING RULES	62
25.2. SETTING PRICING RULES	62
25.3. UPDATE EXISTING PRICING RULES	63
<b>CHAPTER 26. BILLING</b> .....	<b>64</b>
26.1. LISTING INVOICES	64
26.2. INVOICE VIEW	64
26.2.1. Invoice details	64
26.2.2. Editing invoices	65
26.3. INVOICE STATES	65
26.4. AUTOMATED BILLING PROCESS	65
26.4.1. On the first day of each month	65
26.4.2. Every day	66
26.4.3. Automatic and manual invoices	66
26.4.4. Mid-month upgrades	67
26.4.4.1. Prepaid billing	67
26.4.4.2. Postpaid billing	68
26.5. ENABLE/DISABLE BILLING/CHARGING PER ACCOUNT	68
26.6. PLANS WITH A TRIAL PERIOD	69
26.7. VAT RATE/SALES TAX	69
26.7.1. Configure VAT rate field	70

26.7.2. Configure VAT code field	70
26.7.3. Set VAT values for an account	70
26.7.4. Invoices with VAT	70
26.8. DEVELOPER PORTAL VIEW	71
26.9. CREDIT CARD FLOW	71
26.9.1. Sign up for a paid plan	71
26.9.2. Upgrade from a free to a paid plan	72
26.10. "BILLING ADDRESS" FIELD	72
<b>CHAPTER 27. EMAIL NOTIFICATIONS</b>	<b>74</b>
27.1. PROVIDER NOTIFICATIONS	74
27.2. DEVELOPER EMAILS	74
27.2.1. Billing email address	74
<b>CHAPTER 28. BILLING API</b>	<b>76</b>
<b>PART V. SERVICE DISCOVERY: FROM OPENSIFT TO 3SCALE</b>	<b>77</b>
<b>CHAPTER 29. SERVICE DISCOVERY</b>	<b>78</b>
29.1. ABOUT SERVICE DISCOVERY	78
29.2. CRITERIA FOR A DISCOVERABLE SERVICE	78
29.3. CONSIDERATIONS FOR CONFIGURING OPENSIFT TO ENABLE SERVICE DISCOVERY	79
29.4. CONFIGURING SERVICE DISCOVERY WITH AN OPENSIFT OAUTH SERVER	80
29.5. CONFIGURING SERVICE DISCOVERY WITH AN RH-SSO SERVER (KEYCLOAK)	81
29.6. CONFIGURING SERVICE DISCOVERY WITHOUT AN OAUTH SERVER	82
29.7. IMPORTING DISCOVERED SERVICES	83
29.8. AUTHORIZING 3SCALE ACCESS TO AN OPENSIFT PROJECT	84
29.9. UPDATING SERVICES	85
<b>PART VI. MULTITENANCY</b>	<b>86</b>
<b>CHAPTER 30. MULTITENANCY</b>	<b>87</b>
30.1. MASTER ADMIN PORTAL	87
30.1.1. Accessing the Master Admin Portal	87
30.1.2. Adding an account through the Master Admin Portal	87
30.1.3. Creating a single gateway with the Master Admin Portal	88
30.2. MANAGING ACCOUNTS	88
30.2.1. Managing accounts through the Master Admin Portal	88
30.2.2. Managing accounts through API calls	88
30.3. UNDERSTANDING MULTITENANCY SUBDOMAINS	89
30.4. DELETING TENANT ACCOUNTS	90
30.4.1. Deleting an account via the Admin Portal	90
30.4.2. Deleting a tenant via the console	91
30.5. RESUMING TENANT ACCOUNTS	91
<b>PART VII. ANALYTICS</b>	<b>93</b>
<b>CHAPTER 31. IMPLEMENTING 3SCALE API ANALYTICS TO MANAGE AND OPTIMIZE API ACCESS</b>	<b>94</b>
31.1. 3SCALE API METRICS AND METHODS THAT CAPTURE API USE	94
31.2. CONFIGURING 3SCALE PLUGINS TO CAPTURE API METRICS	95
31.3. VIEWING ANALYTICS FOR 3SCALE API BACKENDS	95
31.4. SENDING TEST REQUESTS TO CONFIRM PLUGIN CONFIGURATION FOR CAPTURING API METRICS	95
31.5. TROUBLESHOOTING TECHNIQUES WHEN ANALYTICS ARE MISSING FOR 3SCALE APIS	97
<b>CHAPTER 32. EXPORTING 3SCALE API ANALYTICS BEYOND BUILT-IN CAPABILITIES</b>	<b>98</b>



32.1. EXAMPLE OF USING 3SCALE TO EXTRACT DATA ABOUT APPLICATION USE	98
32.2. EXTRACTING 3SCALE APPLICATION ANALYTICS IN CUSTOM PROCEDURES	98
<b>CHAPTER 33. VIEWING 3SCALE BUILT-IN TRAFFIC ANALYTICS FOR APPLICATIONS</b> .....	<b>102</b>
<b>CHAPTER 34. SETTING UP AND EVALUATING THE 3SCALE RESPONSE CODES LOG FOR YOUR API</b> .	<b>104</b>



## PREFACE

This guide will help you to use the available functions in the Admin Portal to manage your 3scale installation.

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our [CTO Chris Wright's message](#).

## PART I. ACCOUNT SETTINGS

## CHAPTER 1. ACCOUNT CONFIGURATION

After creating your account, update basic information about your company. Set your location and add your contact information.



### NOTE

The account view is only visible to administrators, not to members.

### 1.1. ADD YOUR COMPANY INFORMATION

Once you have created your new account, add your company information with these steps:

1. Click the gear icon located in the right of the top navigation bar. You will see the **Overview** window.
2. Next to the **Account Details** heading, click the *Edit* link.
3. Fill in the information for your account.

The address you specify here has two goals:

- If you are on a paid plan, we use this address for billing purposes.
- If you use the billing and payment modules, this address is also what your user will see on your invoices.

### 1.2. SELECT YOUR PREFERRED TIME ZONE

On the same page you can also select the time zone you will use on all system displays. This setting affects analytics graphs. However, billing cycle calculations are made according to UTC time.

## CHAPTER 2. RED HAT SINGLE SIGN ON FOR THE 3SCALE ADMIN PORTAL

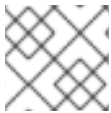
This guide provides information about how to configure and use Red Hat Single Sign On (RH-SSO) with the 3scale Admin Portal.

### 2.1. ENABLE RH-SSO OR AUTH0 MEMBER AUTHENTICATION

3scale supports single sign-on (SSO) authentication for your members and administrators.

The 3scale Admin Portal supports the following SSO providers, each which support a number of identity brokering and member federation options:

- [Red Hat SSO \(RH-SSO\)](#)
- [Auth0](#)



#### NOTE

You can enable multiple SSO member authentication types

Only users that have been added to RH-SSO or Auth0 will be able to access your 3scale Admin Portal through SSO. If you want to further restrict the access by either roles or user groups you should refer to the corresponding step by step tutorials on the [RH-SSO](#) or [Auth0](#) support portals.

Once you have established SSO through your chosen provider, you must configure it and enable it on the 3scale Admin Portal.

#### 2.1.1. RH SSO prerequisites

- An RH SSO instance and realm configured as described under the [Developer Portal authentication](#) section of the documentation.

#### 2.1.2. Auth0 prerequisites

- An Auth0 Subscription and account

#### 2.1.3. Enable RH-SSO

As an administrator, perform the following steps in the 3scale Admin Portal to enable RH-SSO or Auth0:

1. Ensure your preferred SSO provider, highlighted in the prerequisites, has been properly configured
2. Navigate to SSO Integrations in the Account Settings:
  - Click the gear icon in the upper right corner of the page
  - Navigate to **Account Settings (gear icon) > Users > SSO Integrations** and click **New SSO Integration**.
3. Select your SSO provider from the dropdown list
4. Enter the required information, provided when you configured your SSO:

- Client
- Client Secret
- Realm or Site

#### 5. Click **Create Authentication Provider**



#### NOTE

If, during testing, you encounter a callback URL mismatch, add the callback URL shown in the error message to your Auth0 allowed callback URLs.

## 2.2. USING RH-SSO WITH 3SCALE

Once you have configured SSO, members can sign on using the account credentials in connected IdPs.

Follow these steps to log in to the 3scale Admin Portal using SSO:

1. Navigate to your 3scale login page:

```
https://<organization>-admin.3scale.net/p/login
```

2. Authorize 3scale with your IdP
3. If necessary, complete sign up by entering any needed information

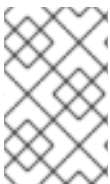
Once you successfully sign up, you will have a member account under the API provider organization, and you will be automatically logged in.

## 2.3. REDIRECTING A 3SCALE LOGIN TO A RH-SSO OPTION

This section describes the redirection to an Identity Provider (IdP) login window via RH-SSO. As a 3scale API Management administrator, complete these steps to have your 3scale account accessible through an optional single sign-on (SSO) login page.

### 2.3.1. Prerequisites

- 3scale 2.11
- An RH-SSO instance and realm configured as described under the [Configuring RH-SSO](#) section of the Developer Portal documentation.



#### NOTE

Before you can integrate RH-SSO with 3scale, you must have a working RH-SSO instance. Refer to the RH-SSO documentation for installation instructions: [Installing RH-SSO 7.2](#).

### 2.3.2. Required steps

1. Access and follow the instructions for setting up RH-SSO under the [Red Hat Single Sign-On for the 3scale Admin Portal](#) section of the 3scale documentation.



2. Provide your RH-SSO administrator with your 3scale URL that will form the basis for a redirect within RH-SSO for your secure logon. Use the following URL format:

`https://<organization>-admin.3scale.net/auth/<system_name>/bounce`

3. **<system\_name>** can be fetched via the SSO Integration detail page of the Admin Portal:

`https://<organization>.3scale.net/p/admin/account/authentication_providers/<ID>`

4. **keycloak\_0123456aaaaa** can also be found via the SSO Integration detail page in the **Callback URL for OAuth flow test** field, which looks like the following:

`https://<organization>.3scale.net/auth/keycloak_0123456aaaaa/callback`

## CHAPTER 3. INVITING USERS AND MANAGING RIGHTS



### NOTE

The *invite* feature is only available for Pro and Enterprise customers.

In order to share the workload of administering your APIs, you may wish to invite team members from your organization to access the 3scale Admin Portal. The instructions below describe how to give access rights to the 3scale Admin Portal, to one or more team members.

With *users*, we refer to the members of your team. The 3scale Admin Portal has two types of users:

- **Admins:** Which have full access to all areas and services, and can invite other members (if your plan allows it).
- **Members:** Which have limited access to areas of the product (e.g. Analytics, Developer Portal) and, if you are an enterprise customer, also to services.

If you create a new 3scale user from a Single Sign-On (SSO) integration, this user has the **member** role by default, regardless of the SSO token content. 3scale does not map its roles to SSO roles.

### 3.1. NAVIGATE TO USER ADMINISTRATION

To see the list of users of your 3scale installation, follow these steps in the Admin Portal page:

1. In the navigation bar, click the gear icon located in the upper right of the window.
2. Navigate to **Users > Listing** from the left side menu.

### 3.2. SEND AN INVITATION

From the list of users, you can invite a new team member. To send the invitation:

1. Click on the *Invite user* link, located on the upper-right side above the list.
2. Enter the email address of the person you want to invite and click **Send**.
3. As a confirmation of the sending, on the upper-right corner of the window you will see a message: **Invitation was successfully sent..**

An invitation email will be sent to the address you entered. If the email does not arrive, make sure it was not marked as spam in the recipient's email account.

Additionally, you can find the list and status of sent invitations in **Users > Invitations**

### 3.3. ACCEPT THE INVITATION

Your new administrator or member must click the link in the invitation email and complete the form to complete the process. Once the form is submitted, their account will be activated.

### 3.4. GIVE NEW USERS RIGHTS

There are two main type of rights you can give to members of your team:

- **By area:** Such as analytics, billing, or developer administration.
- **By service:** Choose which services to give access to members amongst all of your services.  
**Note:** This feature is only available for enterprise customers.

To give a new user rights, edit the new user by selecting them from the user menu and clicking on *Edit*. You have the following user roles:

- Changing their rights to **Admin** will give them full access to control the Admin Portal.
- Changing their rights to **Member** will give you the option of choosing which areas and services the team member has access to.
  - *As Member*, select an area to list all the available services related to said area.

Giving access to certain areas of the Admin Portal will give members access only to the equivalent API:

- Developer accounts – Applications: Gives access to the Account management API
- Analytics: Gives access to the Analytics API
- Billing: Gives access to the Billing API

The screenshot displays the Red Hat 3scale API Management user management interface. Key elements are highlighted with red boxes and arrows:

- Account Settings:** Located in the top navigation bar.
- Users:** A dropdown menu in the left sidebar.
- Role:** A section titled 'ADMINISTRATIVE' with radio buttons for 'Admin (full access)' and 'Member'. An arrow points to this section with the label 'Select a role'.
- This user can access:** A section with checkboxes for 'Developer Portal', 'Billing', 'Settings', 'Accounts -- Applications', 'Analytics', and 'Integration & Application Plans'. An arrow points to this section with the label 'Select areas'.
- Services:** A table with columns for 'Echo API', 'Integration & Application Plans', 'Analytics', and 'Applications'. An arrow points to the first row with the label 'Select services'.

## CHAPTER 4. NOTIFICATIONS

Notifications are sent to admins and members to make it easier to parse developer activity (new account

### 4.1. TYPES OF NOTIFICATIONS

There are different types of notifications:

- Accounts
- Billing
- Applications
- Service subscriptions
- Usage alerts

### 4.2. VISIBILITY

Admin users have access to all notifications.

Member users have access only to notifications of the areas they have been given access to. For example, a member will only have access to notifications related to billing if they have access to the billing section.

For **enterprise accounts**, member users will only have access to notifications regarding activity of the services they have been granted access to.

### 4.3. SUBSCRIBING TO NOTIFICATIONS BY EMAIL

Subscriptions are personal and can only be modified by the person receiving those notifications. To edit your subscriptions:

1. Navigate to **Account Settings (gear icon in the navigation bar) > Personal > Notification Preferences**.
2. Select the notifications you would like to receive.
3. Click **Update Notification Preferences**.

### 4.4. WEB NOTIFICATIONS

In addition to email notifications, you can find information about the last activities in your Dashboard:

The screenshot shows the Red Hat 3scale API Management Admin Portal. The top navigation bar includes the Red Hat logo, the text '3scale API Management', and a 'Dashboard' dropdown menu. Below the navigation bar, the dashboard is divided into several sections. On the left, there is a 'AUDIENCE' section showing '2 Signups last 30 days'. In the center, there is a 'Potential Upgrades' section with a '0 today' indicator and a message: 'Accounts that hit their Usage Limits in the last 30 days. In order to show Potential Upgrades, add 1 or more usage limits to your Application Plans. Furthermore, Web Alerts for Admins of this Account of 100% (and up) should be enabled for service(s) with usage limits.' On the right, there is a 'Web Notifications' section with a red border, containing a list of recent activities: 'TODAY: Running on empty.', 'BEFORE TODAY: curious from Curious Echo signed up', 'Curious Echo has subscribed to your service API', and 'Curious Echo's App created on API'.

## CHAPTER 5. PERSONAL SETTINGS

In Personal settings you can edit your preferences as a team member. If you are an admin, you will also be able to edit the account preferences. For that, check out the [account configuration](#) tutorial.

To edit your personal settings:

1. Click on the gear icon in the navigation bar.
2. On the left panel, navigate to **Personal**. There are 3 types of settings you can edit from here:
  - **Personal Details:** Name, email, password, etc.
  - **Tokens:** Create access tokens to authenticate against the 3scale APIs – Billing, Account Management, and Analytics – and try them out using our ActiveDocs (interactive documentation). Learn more about [3scale tokens](#).
  - **Notification Preferences:** Select which notifications you'd like to receive. Note: If you are an enterprise customer, and if you are a member, these are filtered by area and service. This means you will only be able to subscribe to notifications regarding areas and services you have been given access to. More on [notification preferences here](#).

## CHAPTER 6. TOKENS

This tutorial contains information about 3scale tokens: what are they, how they work, and how to create them.

3scale has two types of tokens: **Access tokens** (created by the user) and **Service tokens** (automatically created when you create a new service in 3scale).

### 6.1. ACCESS TOKENS

Access tokens allow API provider admins and members to authenticate against the 3scale APIs – Billing, Account management, and Analytics – and try them out using our ActiveDocs (interactive documentation).

An access token may provide either read and write access, or read only.

An important thing to take into account is how access tokens work, which is according to the member's rights. Admins can create tokens to authenticate against all three 3scale APIs. Members will be limited by their permissions to access the different parts of the Admin Portal. For example, if a member does not have access to the Billing area, they will not be able to create a token to authenticate against the Billing API.

### 6.2. CREATING ACCESS TOKENS

Access tokens can be created on the tokens page. To create tokens, follow these steps:

1. Click on the gear icon in the navigation bar.
2. Navigate to **Personal > Tokens**
3. Click **Add Access Token**.
4. Specify a name, select one or more scopes, and choose the permission for the token.
5. To save the new token, click **Create Access token**.

Note that if you are a member, you might not see all the APIs – just the ones you have been given access to by the admin of your account.

You can create as many access tokens as you need. For security reasons, the tokens will not be stored on 3scale. When you create a new token, you will be alerted to save the token so you can then use it to make requests to the 3scale API.

If you lose a token, we recommend that you delete it – which will disable it and render it invalid – then create a new one.

### 6.3. USING ACCESS TOKENS

When using your access token to make calls to the 3scale APIs the results will be filtered by the services you have access to.

For example, when deploying APIcast self-managed, you'll need an access token so your APIcast API gateway can pull the configuration of the service using the Account Management API.

The way it works is if your organization has set up three services on 3scale, and as a member, you have

access to Service 1, but not 2 and 3, and you also have access to the Account Management API, when you create a token and make a request to the Account Management API you will only get the applications which are using Service 1.

Following the same example, if you have access to the Account Management API, but access to zero services, when making a call, you'll get "access denied" error.

### 6.3.1. Service tokens

Service tokens are used to authenticate against 3scale Service Management API. Service tokens are generated automatically when a new service is created in 3scale, and are unique per service. They are shared among the users of the 3scale account.

You can find the service tokens for the services that the user has access to in the Admin Portal's Dashboard: **Account Settings (gear icon) > Personal > Tokens**

## PART II. MANAGING DEVELOPER ACCOUNTS



## CHAPTER 7. ADDING DEVELOPERS

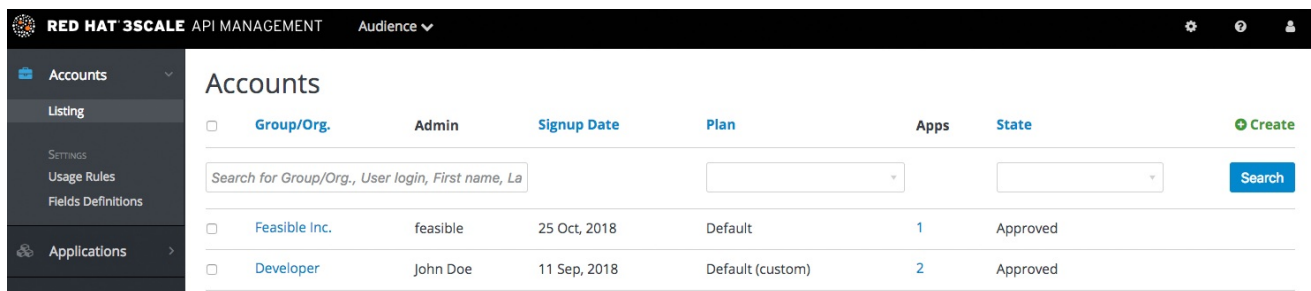
These are the steps to add a new developer account for access to your API.

If you have configured the workflow to invite developers manually, this covers how to add new developers.

### 7.1. CREATE A NEW DEVELOPER ACCOUNT

1. Follow *Accounts* link from the **Audience** section on the Admin Portal.
2. Click **Create**.

As an admin, you can skip even some of the required fields. If you want to invite users to the account securely, you can also skip the password fields. However the email on this main admin account must be unique among all users.

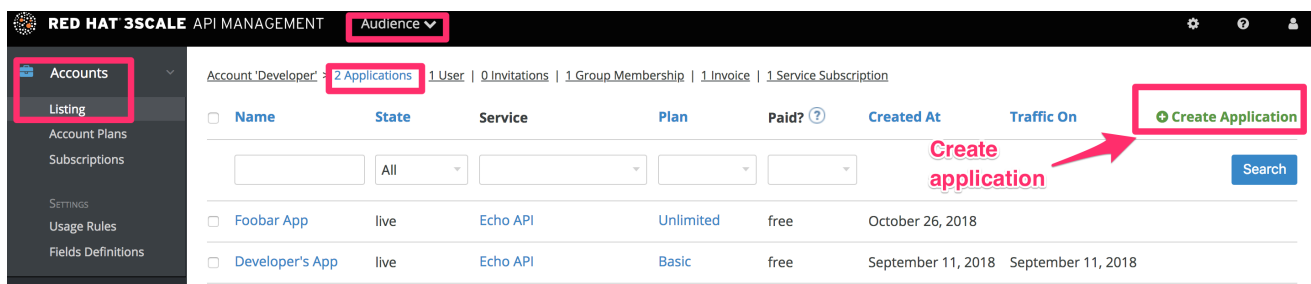


The screenshot shows the 'Accounts' page in the Red Hat 3Scale API Management interface. The page title is 'Accounts' and it is under the 'Audience' section. There is a search bar and a 'Create' button. The table below shows the following data:

Group/Org.	Admin	Signup Date	Plan	Apps	State
Feasible Inc.	feasible	25 Oct, 2018	Default	1	Approved
Developer	John Doe	11 Sep, 2018	Default (custom)	2	Approved

### 7.2. SET UP APPLICATIONS

If you want to pre-configure app keys for the account, you can also add an application on behalf of the developer. Otherwise, leave this as one of the initial steps for the developer to take.



The screenshot shows the 'Applications' page in the Red Hat 3Scale API Management interface. The page title is 'Applications' and it is under the 'Audience' section. There is a search bar and a 'Create Application' button. The table below shows the following data:

Name	State	Service	Plan	Paid?	Created At	Traffic On
Fooobar App	live	Echo API	Unlimited	free	October 26, 2018	
Developer's App	live	Echo API	Basic	free	September 11, 2018	September 11, 2018

### 7.3. NOTIFY THE DEVELOPER

You can either send an email invitation to the developer manually or follow the steps to use the [invite developer](#) feature.

## CHAPTER 8. APPROVING DEVELOPERS

This section shows how to make approvals for any step in the signup workflow.

Once you've implemented the signup workflow with manual approval steps, you have a few options. The approval process is slightly different depending on the trigger and what is being approved. If you receive an email notification, follow the instructions in the following section. Otherwise, it depends on whether you want to approve an account, a service, or an application.

### 8.1. APPROVE FROM EMAIL NOTIFICATION

If you (as admin) receive an email notification that one of your developers has an item pending approval, you can copy/paste the URL for the item into your browser, and it will take you directly to the page to make the approval.

### 8.2. ACCOUNT APPROVAL

To search for specific accounts or filter all accounts that are in a "pending" (for approval) state, navigate to **Audience > Accounts > Listing**. To show only the pending accounts, select "Pending" in the dropdown list *State* and click **Search**.

You can make individual approvals directly on each row, or select several rows at a time and perform a bulk approval.

**RED HAT 3SCALE** API MANAGEMENT Audience ▾

**Accounts**

**Bulk operations**

You have selected 1 accounts and you can make following operations with them:

**Select multiple accounts for bulk approval**

Send email to selected accounts

Transfer these accounts to different account plan

Approve, reject or make pending selected accounts

**Filter for accounts "pending" approval**

Change state

Search Group/Org., User login, First name, Last

	Group/Org.	Admin	Signup Date	Plan	Apps	State	Create
<input checked="" type="checkbox"/>	tester inc	tester	25 Oct, 2018	Default	1	Created	Activate
<input type="checkbox"/>	barba papa	barbapapa	25 Oct, 2018	Default	1	Created	Activate
<input type="checkbox"/>	Feasible Inc.	feasible	25 Oct, 2018	Default	1	Approved	
<input type="checkbox"/>	Developer	John Doe	11 Sep, 2018	Default (custom)	2	Approved	

### 8.3. SERVICE APPROVAL

To search for specific subscriptions to a service or filter all subscriptions that are in a "pending" (for approval) state, navigate to **Audience > Accounts > Subscriptions**

To view Subscriptions, enable Service Plans in **Audience > Accounts > Usage Rules**

You can select one subscription or several at a time and perform a bulk approval.

**RED HAT 3SCALE** API MANAGEMENT Audience

## Accounts

**Bulk operations**

You have selected 1 accounts and you can make following operations with them:

- Send email: Send email to selected accounts
- Change account plan: Transfer these accounts to different account plan
- Change state**: Approve, reject or make pending selected accounts

Search for Group/Org., User login, First name, L

<input type="checkbox"/>	Group/Org.	Admin	Signup Date	Plan	Apps	State	
<input checked="" type="checkbox"/>	Developer	John Doe	11 Sep, 2018	Default (custom)	3	<b>Pending</b>	Approve

[Export all Accounts](#)

## 8.4. APPLICATION APPROVAL

To search for applications or filter all applications that are in a “pending” (for approval) state:

1. Navigate to **Audience > Applications > Listing**
2. Select "pending" in the dropdown list *State* and click **Search**.

You can select one application or several at a time and perform a bulk approval.

**RED HAT 3SCALE** API MANAGEMENT Audience

## Applications

**Bulk operations**

You have selected 1 applications and you can make following operations with them:

- Send email: Send email to owners of selected applications
- Change application plan: Transfer these applications to different application plan
- Change state**: Accept, suspend or resume selected applications

**Perform bulk changes**

**Filter for the required state**

<input type="checkbox"/>	Name	State	Account	Service	Plan	Paid? ⓘ	Created At	Traffic On
<input checked="" type="checkbox"/>	barba papa's App	live	barba papa	Echo API	Unlimited	free	October 25, 2018	
<input type="checkbox"/>	Feasible Inc.'s App	live	Feasible Inc.	Echo API	Basic	free	October 25, 2018	
<input type="checkbox"/>	Developer's camel-rest-sql App	live	Developer	camel-rest-sql	Default	free	October 11, 2018	
<input type="checkbox"/>	Developer's App	live	Developer	Echo API	Basic	free	September 11, 2018	September 11, 2018

You can also start from the details page for a developer account, select which application you wish to approve from there, and make the approval on the application details page.

**RED HAT 3SCALE** API MANAGEMENT Audience ▾

**Accounts** ▾

- Listing
- Account Plans
- Subscriptions

SETTINGS

- Usage Rules
- Fields Definitions

**Applications** >

**Billing** >

**Account 'Developer'** 3 Applications | 1 User | 0 Invitations | 1 Group Membership | 1 Invoice

## Account: Developer [Edit](#)

Organization/Group Name	Developer	<a href="#">Send message</a>
Administrator	John Doe ( <a href="mailto:admin+test@3scale.poc-sd.staging.3sca.net">admin+test@3scale.poc-sd.staging.3sca.net</a> )	
Signed up on	September 11, 2018 13:22	
Status	Pending	<a href="#">Approve</a> <a href="#">Reject</a>

**RED HAT 3SCALE** API MANAGEMENT API: Echo API ▾

Overview

Analytics >

**Applications** ▾

- Listing
- Application Plans

Subscriptions >

ActiveDocs

Integration >

**Application 'My app'** [Analytics](#)

## My app [Edit](#)

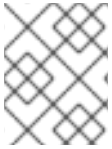
Account	Developer	
Description	my app	
Service	Echo API	
State	Pending	<a href="#">Accept</a> or <a href="#">Reject</a>

### API Credentials

## CHAPTER 9. CHANGING PLANS FOR AN APP

After this section you will be able to change plans for accounts, services or applications.

As admin you may change plans for a developer at any time, or in response to a plan change request that the developer initiates.



### NOTE

The change plans step is slightly different depending on what type of plans are being changed.

### 9.1. CHANGE ACCOUNT PLANS

To search or filter specific accounts, navigate to **Audience > Accounts > Listing**

You can select one or more rows at a time, and change the plans.

**RED HAT 3SCALE** API MANAGEMENT **Audience** ▾

**Accounts**

**Bulk operations**

You have selected 2 accounts and you can make following operations with them:

Send email to selected accounts

Transfer these accounts to different account plan

Approve, reject or make pending selected accounts

<input type="checkbox"/>	Group/Org.	Admin	Signup Date	Plan	Apps
<input checked="" type="checkbox"/>	Feasible Inc.	feasible	25 Oct, 2018	Default	1
<input checked="" type="checkbox"/>	Developer	John Doe	11 Sep, 2018	Default (custom)	2

### 9.2. CHANGE SERVICE PLANS

To search or filter specific subscriptions to a service, navigate to **Audience > Accounts > Subscriptions**

You can only view subscriptions if you have enabled Service Plans in the Settings page.

You can select one or several subscriptions at a time, and change the plans.

**RED HAT 3SCALE** API MANAGEMENT Audience

## Service Subscriptions

**Bulk operations**

You have selected 1 service subscriptions and you can make following operations with them:

Send email Send email to selected subscribers

**Change service plan** Transfer these subscriptions to different service plan

Change state Accept, suspend or resume selected subscriptions

<input type="checkbox"/>	Account	Service	Plan	State	Paid? <sup>?</sup>	Created On
<input type="checkbox"/>						
<input type="checkbox"/>	barba papa	Echo API	Default	live	free	October 25, 2018
<input checked="" type="checkbox"/>	Feasible Inc.	Echo API	Default	live	free	October 25, 2018
<input type="checkbox"/>	Developer	camel-rest-sql	Default	live	free	October 11, 2018
<input type="checkbox"/>	Developer	Echo API	Default	live	free	September 11, 2018

## 9.3. CHANGE APPLICATION PLANS

To search or filter specific applications, navigate to **Audience > Applications > Listing**

You can select one or several applications at a time, and change plans.

**RED HAT 3SCALE** API MANAGEMENT Audience

## Applications

**Bulk operations**

You have selected 2 applications and you can make following operations with them:

Send email Send email to owners of selected applications

**Change application plan** Transfer these applications to different application plan

Change state Accept, suspend or resume selected applications

<input type="checkbox"/>	Name	State	Account	Service	Plan	Paid? <sup>?</sup>	Created At
<input type="checkbox"/>		All					
<input checked="" type="checkbox"/>	Feasible Inc.'s App	live	Feasible Inc.	Echo API	Basic	free	October 25, 2018
<input type="checkbox"/>	Developer's camel-rest-sql App	live	Developer	camel-rest-sql	Default	free	October 11, 2018
<input checked="" type="checkbox"/>	Developer's App	live	Developer	Echo API	Basic	free	September 11, 2018

Another scenario is to start from the details page for a developer Account. From there you select the application for which you wish to change plan. On the application details page, you can change the plan.

The screenshot displays the Red Hat 3Scale API Management interface. The top navigation bar includes the Red Hat 3Scale logo, 'API MANAGEMENT', and a dropdown menu for 'API: Echo API'. The left sidebar contains navigation options: Overview, Analytics, Applications (highlighted), Listing, Application Plans, Subscriptions, ActiveDocs, and Integration. The main content area is titled 'Metro's App' and includes an 'Edit' link. It features a table with application details: Account (Metro), Description (Default application created on signup), Service (Echo API), and State (Live with a 'Suspend' button). Below this is the 'API Credentials' section, showing a 'User Key' (masked as xxxxxxxx) and a 'Regenerate' button. To the right, the 'Application Plan: Unlimited' is shown with a 'Convert to a Custom Plan' option and a list of features: Unlimited Greetings, 24/7 support, and Unlimited calls, each with a status indicator. At the bottom right, a 'Change Plan' dropdown menu is open, showing options for Basic, Plan A (selected), and Plan B.

### 9.3.1. More information

If rather than change to another standard plan, you only want to make a change for one specific app, you can use the [customize plans](#) feature.

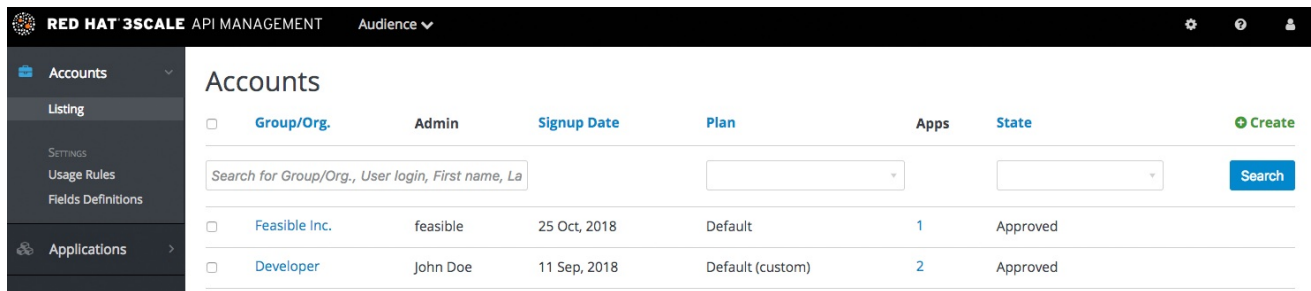
## CHAPTER 10. CONTACTING DEVELOPERS

This guide explains how to find out which developer account manages a particular application and then communicate with them – both through 3scale and externally.

During API operations, you may urgently need to contact developers who are using your API.

### 10.1. LOCATE THE RELEVANT APPLICATION AND ACCOUNT IN THE SYSTEM

If you already know the account and developer who manages the application in question, navigate to their account from the Accounts page in **Audience > Accounts > Listing** as shown below.



The screenshot shows the Red Hat 3scale API Management Admin Portal. The top navigation bar includes the Red Hat 3scale logo, 'API MANAGEMENT', and the user 'Audience'. A sidebar on the left contains navigation options: 'Accounts' (selected), 'Listing', 'SETTINGS' (Usage Rules, Fields Definitions), and 'Applications'. The main content area is titled 'Accounts' and features a table with columns: Group/Org., Admin, Signup Date, Plan, Apps, and State. A search bar is located above the table, and a 'Create' button is in the top right. The table lists two accounts: 'Feasible Inc.' and 'Developer'.

<input type="checkbox"/>	Group/Org.	Admin	Signup Date	Plan	Apps	State	<a href="#">Create</a>
<input type="checkbox"/>	<input type="text" value="Search for Group/Org., User login, First name, La"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Search"/>
<input type="checkbox"/>	<a href="#">Feasible Inc.</a>	feasible	25 Oct, 2018	Default	1	Approved	
<input type="checkbox"/>	<a href="#">Developer</a>	John Doe	11 Sep, 2018	Default (custom)	2	Approved	

If you only have the application ID or API key, you can use the search box on the Accounts page in **Audience > Accounts > Listing** to find the relevant account. More information on locating applications is available [here](#).

### 10.2. SEND INTERNAL MESSAGES TO DEVELOPERS

Once you are in the account profile page as shown below, click the message icon.



**RED HAT 3SCALE** API MANAGEMENT Audience ▾

Account 'Developer' > [2 Applications](#) | [1 User](#) | [0 Invitations](#) | [1 Group Membership](#) | [1 Invo](#)

## Account: Developer [Edit](#)

Organization/Group Name	Developer	<a href="#">Send message</a>
Administrator	John Doe ( <a href="mailto:admin+test@3scale.poc-sd.staging.3sca.net">admin+test@3scale.poc-sd.staging.3sca.net</a> )	
Signed up on	September 11, 2018 13:22	
Status	Approved	

### Billing Status

✔ Monthly billing is enabled. ⏹ Disable

The message created here will be sent both to the account system dashboard, where all developers on the account will see it, and by email to the people on the developer account who have admin status within the account.

### 10.3. CONTACT BY OTHER MEANS

If it's an emergency and email is unlikely to be fast enough for your purposes, you can also use the contact information submitted by the developer at time of signup, which is available:

- On the company account page (general contact information but may include a phone number)
- Developer/user specific information on the users' own file

Note that you can make contact phone numbers a required field upon signup.

## CHAPTER 11. CUSTOMIZE PLANS

When you have completed this section you will have customized an application plan for a specific developer.

Application plans are a good way to apply standard policies for different segments of your developer community. However, you always have the flexibility to customize the standard plans for any individual developer with unique requirements.

Once a plan is customized, you lose the link to the original plan. If you make changes to the original plan, the custom plan does not inherit any of those changes. So you should use this customization feature sparingly, before you become overwhelmed with too many custom plans which you cannot manage effectively.

A developer wants to increase their current limits without upgrading to the next pricing tier as the current billing period is already under way. A customization could be a good way to handle this situation by enabling the increase in limits and charging only the variable costs incurred. This would also help encourage an upgrade for the following billing month.

### 11.1. CHOOSE THE ACCOUNT

To view the details page for the developer Account you are interested in:

1. Navigate to **Audience > Accounts > Listing**
2. Choose the Developer account

RED HAT 3SCALE API MANAGEMENT Audience

Accounts

<input type="checkbox"/>	Group/Org.	Admin	Signup Date	Plan	Apps	State	<a href="#">+ Create</a>
<input type="checkbox"/>	Feasible Inc.	feasible	25 Oct, 2018	Default	1	Approved	
<input type="checkbox"/>	Developer	John Doe	11 Sep, 2018	Default (custom)	2	Approved	

### 11.2. SELECT THE APPLICATION

Select the application whose plan you wish to customize.

RED HAT 3SCALE API MANAGEMENT Audience

Account 'Developer' 2 Applications | 1 User | 0 Invitations | 1 Group Membership | 1 Invoice | 1 Service Subscription

<input type="checkbox"/>	Name	State	Service	Plan	Paid? <a href="#">?</a>	Created At	Traffic On	<a href="#">+ Create Application</a>
<input type="checkbox"/>	Foobar App	live	Echo API	Unlimited	free	October 26, 2018		
<input type="checkbox"/>	Developer's App	live	Echo API	Basic	free	September 11, 2018	September 11, 2018	

### 11.3. CUSTOMIZE THE APPLICATION PLAN

Select the option to “customize”. This provides the page where all the plan elements can be customized for the application owned by this account.

The screenshot displays the Red Hat 3Scale API Management interface for an application named 'Metro's App'. The left sidebar contains navigation options: Overview, Analytics, Applications (highlighted), Listing, Application Plans, Subscriptions, ActiveDocs, and Integration. The main content area is titled 'Metro's App' and includes an 'Edit' link. Below the title is a table with the following details:

Account	Metro
Description	Default application created on signup.
Service	Echo API
State	Live <span style="color: red;">● Suspend</span>

Below the table are three sections:

- API Credentials:** Shows a 'User Key' with a 'Regenerate' button.
- Usage in last 30 Days:** A chart showing 'Hits' with a value of '0 hits'.
- Current Utilization:** A message stating 'This is an unmetered application, there are no limits defined'.

On the right side, there is a section for 'Application Plan: Unlimited' with a 'Convert to a Custom Plan' button. Below this is a 'FEATURES' table:

Unlimited Greetings	<span style="color: green;">✔</span>
24/7 support	<span style="color: green;">✔</span>
Unlimited calls	<span style="color: green;">✔</span>

At the bottom right, there is a 'Change Plan' section with a dropdown menu and a 'Change Plan' button.

### 11.3.1. More information

Before you take the step to customize plans, always consider first if you are not better off with a new standard plan (which can be hidden from display in the Developer Portal). Then you would just [change to the standard plan](#) and so gain the benefit of reuse if this applies to more than one of your developer partners.

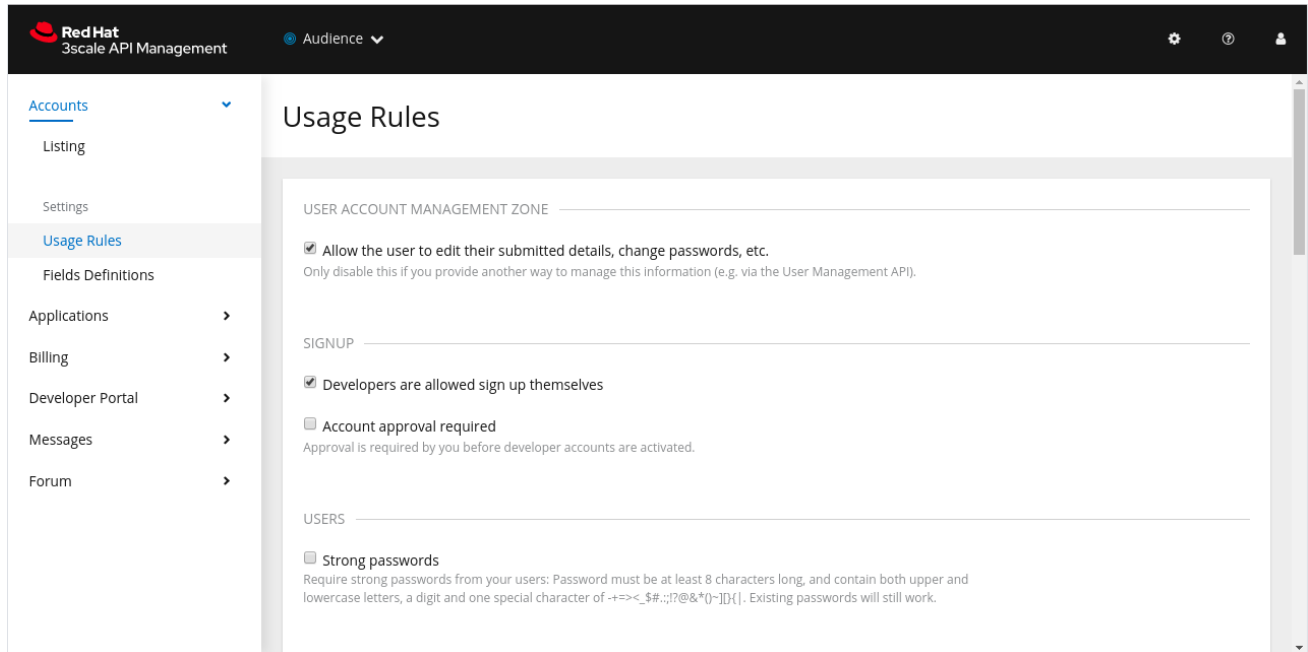
## CHAPTER 12. ENABLE SIGNUP

Configure developer signup by implementing self-service or manual mode.

You can configure the workflow for developers to be self-service or by manual invite only. Self-service signups are done by developers through the Developer Portal, while manual invites are handled by your admins through the Admin Portal.

By default, developers are enabled to sign up by themselves. If you enable developer self-service, you can require admin approval before the developer account can be activated.

To do so, navigate to **Audience > Accounts > Settings > Usage Rules**



The screenshot displays the Red Hat 3scale API Management Admin Portal interface. The top navigation bar includes the Red Hat logo, the text "Red Hat 3scale API Management", a dropdown menu for "Audience", and icons for settings, help, and user profile. The left sidebar contains a navigation menu with the following items: "Accounts" (expanded), "Listing", "Settings", "Usage Rules" (selected), "Fields Definitions", "Applications", "Billing", "Developer Portal", "Messages", and "Forum". The main content area is titled "Usage Rules" and contains three sections:

- USER ACCOUNT MANAGEMENT ZONE**:
  - Allow the user to edit their submitted details, change passwords, etc.**  
Only disable this if you provide another way to manage this information (e.g. via the User Management API).
- SIGNUP**:
  - Developers are allowed sign up themselves**
  - Account approval required**  
Approval is required by you before developer accounts are activated.
- USERS**:
  - Strong passwords**  
Require strong passwords from your users: Password must be at least 8 characters long, and contain both upper and lowercase letters, a digit and one special character of `+><_$.:;!@&*()-|[]{}.` Existing passwords will still work.

## CHAPTER 13. FINDING APPLICATIONS

By the end of this guide, you'll be able to quickly locate an application in the Admin Portal based on either its name, an API key, or an application identifier.

During API operations, you may need to be able to find information on an application that is accessing your API quickly – either for support purposes, to change configuration, or potentially because the application is misbehaving and needs to be disabled.

### 13.1. GET THE INFORMATION YOU NEED

To find an application, you need the name of the account it belongs to or the application's name. If you do not have this information, you can verify the access logs. To perform the search, navigate to Applications (**Audience > Applications > Listing**).

If you search by identifier for an authentication type, you need the following information:

- For API key-only authentication patterns: the API key
- For app ID and app key authentication patterns: the app identifier (search by app key is not supported)
- For OpenID Connect authentication patterns: the client\_id (search on the secret is not supported)

### 13.2. SEARCH FOR THE APPLICATION

To search a given application, navigate to Applications page (**Audience > Applications > Listing**), and use the search box as shown in the image below.

The screenshot shows the Red Hat 3Scale API Management interface. The top navigation bar includes the Red Hat logo, 'RED HAT 3SCALE API MANAGEMENT', and a dropdown menu for 'Audience'. A left sidebar contains navigation items: Accounts, Applications (highlighted with a red box), Listing, Alerts, Billing, Portal, and Messages. The main content area is titled 'Applications' and features a search bar and a table of applications.

<input type="checkbox"/>	Name	State	Account	Service
<input type="checkbox"/>	Feasible Inc.'s App	live	Feasible Inc.	Echo API
<input type="checkbox"/>	Developer's camel-rest-sql App	live	Developer	camel-rest-sql
<input type="checkbox"/>	Developer's App	live	Developer	Echo API

### 13.3. ACCESS APPLICATION INFORMATION

Once the results are returned, click on the application you'd like to access and you'll be taken to that application's homepage, which includes information such as that shown in the image below.

**RED HAT 3SCALE** API MANAGEMENT Audience ▾

- Accounts >
- Applications** ▾
  - Listing
  - Alerts
- Billing >
- Portal >
- Messages >

## Applications

<input type="checkbox"/> Name	State	Account	Service
<input type="checkbox"/> Feasible Inc.'s App	live	Feasible Inc.	Echo API
<input type="checkbox"/> Developer's camel rest-sql App	live	Developer	camel-rest-sql
<input type="checkbox"/> Developer's App	live	Developer	Echo API

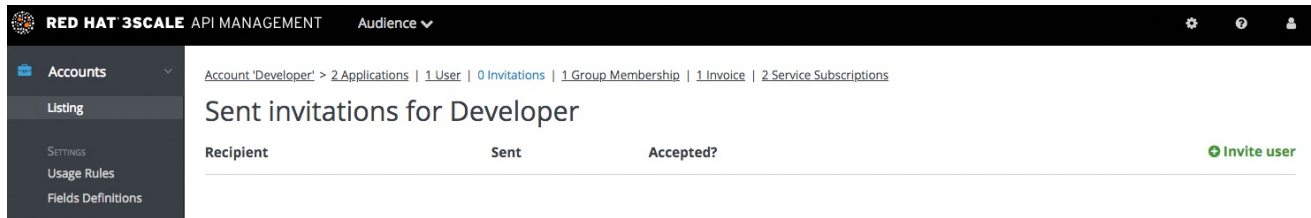
**Select the App to check**

## CHAPTER 14. INVITING DEVELOPERS

After completing these steps, you will have added a new developer user to a developer account.

When you create a developer account manually, you can invite developer users to that account through the Admin Portal:

1. Navigate to **Audience > Accounts > Listing**
2. Choose the account in question.
3. Select "Invitations" and then click **Invite user**.



The screenshot shows the Red Hat 3Scale API Management Admin Portal. The top navigation bar includes the Red Hat 3Scale logo, 'API MANAGEMENT', and a dropdown menu for 'Audience'. The left sidebar contains a menu with 'Accounts' (selected), 'Listing', and 'SETTINGS' (with sub-items 'Usage Rules' and 'Fields Definitions'). The main content area shows the breadcrumb 'Account 'Developer' > 2 Applications | 1 User | 0 Invitations | 1 Group Membership | 1 Invoice | 2 Service Subscriptions' and the title 'Sent invitations for Developer'. Below the title is a table with columns 'Recipient', 'Sent', and 'Accepted?'. A green 'Invite user' button is located in the top right corner of the table area.

## CHAPTER 15. UNSUBSCRIBING DEVELOPERS FROM A SERVICE

As an admin, you can unsubscribe developers from a service. You may need to do this for one specific developer, or for multiple developers, in the event of a service deprecation.

### 15.1. UNSUBSCRIBING A SINGLE DEVELOPER FROM SERVICES

Unsubscribe a single developer from a service they are subscribed to through the Admin Portal:

1. In the Admin Portal's Dashboard, navigate to **Audience > Accounts > Listing > [select an account] > Service Subscriptions**.
2. Select **Unsubscribe** for the service that you want to remove the developer from.

### 15.2. UNSUBSCRIBING MULTIPLE DEVELOPERS FROM SERVICES

Perform a bulk action to unsubscribe multiple developers from a deprecated or deleted service:



#### NOTE

This method only applies to services that have been deleted or suspended. You cannot perform a bulk unsubscription action on active services.

1. In the Dashboard, navigate to: **Audience > Accounts > Subscriptions**
2. Do bulk state change.
3. Using the service dropdown menu, identify the service from which you want to unsubscribe developers.
4. Using the checkboxes on the left, select the developers you want to unsubscribe.
5. Select **Change State > Suspend** to suspend the selected developer subscriptions.

Remember that service plans need to be enabled.



## CHAPTER 16. SUSPENDING APPLICATIONS

This guide explains how to disable all keys and access tokens for an application.

If an application is misusing your API and affecting other traffic, you may need to quickly suspend its operations before contacting the developer involved to ask them to amend their code or configuration.

### 16.1. FIND THE APPLICATION

You can find the application from the **Accounts** or **Applications** tabs or by searching as described [here](#).

### 16.2. DISABLE THE APPLICATION

Once you have located the application and see the application summary page, click on the suspend icon next to the 'State' value. This action will immediately disable the application from the API and suspend all keys from working. Calls with these application keys will be rejected by the control system.

The application can be unsuspended using the same button once the problematic behavior has been rectified.

The screenshot displays the Red Hat 3Scale API Management interface. The top navigation bar shows 'RED HAT 3SCALE API MANAGEMENT' and 'API: Echo API'. The left sidebar contains navigation links: Overview, Analytics, Applications (selected), Listing, Application Plans, ActiveDocs, and Integration. The main content area is titled 'Application 'Developer's App' | Analytics' and 'Developer's App' with an 'Edit' link. Below this is a table of application details:

Account	Developer
Description	Default application created on signup.
Service	Echo API
State	Live <span style="color: red;">⏸ Suspend</span>

Below the table is a section for 'API Credentials'.




#### NOTE

If you use caching in your agents, suspension may not be immediate but require a short timeout.

### 16.3. CONTACT THE DEVELOPER

How you contact the developer of the application will depend on your workflow and policy. On the same page, you can click on the account name, which will take you to the account view where you can identify the key administrator of the account that owns the application. You can contact them either by email or by clicking on the send message button as shown, which will generate a dashboard message for the user.

 **RED HAT 3SCALE** API MANAGEMENT Audience ▾

- Accounts ▾
  - Listing
  - SETTINGS
    - Usage Rules
    - Fields Definitions
- Applications >
- Billing >
- Portal >
- Messages >

Account 'Developer' > [2 Applications](#) | [1 User](#) | [0 Invitations](#) | [1 Group Membership](#) | [1 Invo](#)

## Account: Developer [Edit](#)

Organization/Group Name	Developer	<a href="#">Send message</a>
Administrator	John Doe ( <a href="mailto:admin+test@3scale.poc-sd.staging.3sca.net">admin+test@3scale.poc-sd.staging.3sca.net</a> )	
Signed up on	September 11, 2018 13:22	
Status	Approved	

### Billing Status

✔ Monthly billing is enabled. [Disable](#)

## CHAPTER 17. DELETING APPLICATIONS

To delete an application via the Admin Portal, you need to follow these steps:

**Option 1:** Delete an application from the list of all applications for [Your\_API\_name].

1. In the Dashboard, click [Your\_API\_name].
2. Click the **Overview** tab.
3. From the left panel on the *Overview* page, click **Applications**.
4. Choose **Listing**.
5. Click on an application.
6. You will see a page containing details of the application. Click **Edit**.
7. To delete the application, click **Delete**.
8. You will see a confirmation message. Click **Ok** to confirm the deletion.

**Option 2:** Delete an application based on a specific application plan.

1. In the Admin Portal, click **Dashboard**.
2. Choose **API**.
3. Under *Published Application Plans*, choose an application.
4. Click on an application.
5. You will see a page containing details of the application. Click **Edit**.
6. To delete the application, click **Delete**.
7. You will see a confirmation message. Click **Ok** to confirm the deletion.

Alternatively, you can also delete an application via 3scale API Docs, with the operation called **Application Delete**.

## CHAPTER 18. DELETING AN API

You can delete an API by deleting its service. Deleting a service removes the API's applications, application plans, metrics, pricing rules, features, service plans, and subscriptions.

To delete an API:

1. Navigate to **[Your\_product\_name] > Overview > Edit**
2. In the Service deletion section, click the link to delete the service.

## PART III. ACCESS CONTROL

As a 3scale API provider, you control access to your API by designating methods for which to capture individual usage details, defining metrics, associating methods and metrics with mapping rules, and creating application plans that set prices and usage limits. Applications plans use data collected for methods and metrics to enforce usage limits.

## CHAPTER 19. DESIGNATING METHODS AND ADDING METRICS FOR CAPTURING USAGE DETAILS

An application plan sets limits and pricing rules for consumer access to your API. To enable enforcement of limits and rules, designate methods in your API for which to collect individual usage data or add metrics. Add a mapping rule to each designated method and each custom metric. The mapping rule specifies details about the usage data that you want to capture.

You can designate methods or add metrics for products and backends. For products, this enables setting limits and pricing rules in the product's application plans. For backends, this enables setting limits and pricing rules in the application plans for any product that bundles that backend.

Designate methods for which to capture the number of individual calls. This provides finer granularity for tracking API use. Reporting traffic to a method automatically increases counters for the method and for the *Hits* metric. You can designate a method for each endpoint of your API backend or for a combination of endpoint and HTTP method. See [Adding mapping rules to methods and metrics](#) to learn how to map the endpoints of your API to the methods added here.

Metrics are suitable to track the usage of your API, for both products and backends. *Hits* is the built-in metric that exists for each API. It tracks the number of calls made to your API. To capture usage of your API apart from *Hits*, define a metric that reports the usage in different units. A unit should be quantifiable and apply a meaning for your business goals such as megabytes, CPU time, or the number of elements returned by the API. Metrics other than *Hits*, such as CPU time or **mb** are not provided by default. Obtain these metrics by using an endpoint called by an external service configured by the user.

Methods and metrics are the scaffolding for packaging your API. Each application plan enables you to define different usage limits and pricing rules for each designated method and each metric. See [API analytics](#) to learn more about the usage reported by metrics and methods.

### 19.1. ADDING METHODS TO PRODUCTS AND BACKENDS

Adding a method to a product or backend means that you are designating a method in your API for which you want to capture individual usage details. An application plan provides the ability to set a limit for each method that you add to a product or backend. The procedure for adding a method or metric to a product is similar to adding a method or metric to a backend.

#### Procedure

1. Navigate to **[Your\_product\_name] > Integration > Methods & Metrics** or **[Your\_backend\_name] > Methods & Metrics**
2. Click *New method*.
3. In the **Friendly name** field, enter a short description of the method. This name is displayed in different sections of the 3scale Admin Portal. The friendly name must be unique for the product.



#### IMPORTANT

Be careful with changing the system name of the methods or deleting them. These changes can break your already deployed 3scale integration if there are mapping rules pointing to the previous system name of the method.

4. In the **System name** field, enter the name of the method in your API to use to report the usage through the 3scale Service Management API.  
You decide the format of the system name. This name can be the similar to the endpoint (**/status**), or, for example, it can include the method and the path ( **GET\_/status**). However, the system name must conform to these rules:
  - Unique in the product or backend
  - Contain only alphanumeric characters, underscore `_`, hyphen `-` or forward slash `/`
  - No spaces
  - For backend APIs, the system name of methods includes a numeric string that identifies the backend they are mapped to. You cannot modify this backend identifier.
5. Optional: In the **Description** field, enter a more detailed description of the method.
6. Click **Create Method**.

### Verification steps

- Added methods are available in your application plans.

### Next steps

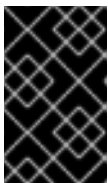
- Edit limits and pricing rules for each method by going to **[Your\_product\_name] > Applications > Application Plans > [plan\_you\_want\_to\_edit]**.

## 19.2. ADDING METRICS TO PRODUCTS AND BACKENDS

Adding a metric specifies a usage unit that you want to capture for all calls to your API. An application plan provides the ability to set a limit for each metric that you add to a product or backend. The procedure for adding a method or metric to a product is similar to adding a method or metric to a backend.

### Procedure

1. Navigate to **[Your\_product\_name] > Integration > Methods & Metrics** or **[Your\_backend\_name] > Methods & Metrics**
2. Click *New metric*.
3. In the **Friendly name** field, enter a short description of the metric. This name is displayed in different sections of the 3scale Admin Portal. The friendly name must be unique for the product.



### IMPORTANT

Be careful with changing the system name of the metrics or deleting them. These changes can break your already deployed 3scale integration if there are mapping rules pointing to the previous system name of the metric.

4. In the **System name** field, enter the name of the metric in your API to use to report the usage through the 3scale Service Management API.

You decide the format of the system name. However, the system name must conform to these rules:

- Unique in the product or backend
  - Contain only alphanumeric characters, underscore `_`, hyphen `-` or forward slash `/`
  - No spaces
  - For backend APIs, the system name of metrics includes a numeric string that identifies the backend they are mapped to. You cannot modify this backend identifier.
5. In the **Unit** field, enter the unit.
    - Use a singular noun, for example, *hit*. The singular will become plural in the analytics charts.
  6. Optional: In the **Description** field, enter a more detailed description of the metric.
  7. Click **Create Metric**

#### Verification steps

- Added metrics are available in your application plans.

#### Next steps

- Edit limits and pricing rules for each metric by going to `[Your_product_name] > Applications > Application Plans > [plan_you_want_to_edit]`.
- Map your metrics to one or more URL pattern by going to `[Your_product_name] > Integration > Mapping Rules`. See [Adding mapping rules to methods and metrics](#).

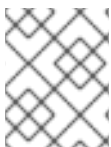
## 19.3. ALTERNATIVES FOR IMPORTING METHODS AND METRICS

If your API has multiple endpoints, there are two ways to automatically designate methods and add metrics to 3scale products and backends:

- [Importing via Swagger spec](#).
- [Importing via RAML spec](#).

## 19.4. ADDING MAPPING RULES TO METHODS AND METRICS

Mapping rules are operations that are mapped to previously created [methods](#) and [metrics](#) in your products and backends.



#### NOTE

Mapping rules are required in your previously created methods, however, they are optional for metrics.

#### Procedure

1. Navigate to `[Your_product_name] > Integration > Mapping Rules`



2. Click **Create mapping rule**.
3. The **Verb** field is pre-populated with the HTTP method, **GET**, however you can select other options from the dropdown list.
4. In the **Pattern** field, add a valid URL that starts with an forward slash `/`. The URL can be from a wildcard you specified inside curly brackets `{}`.
5. In the **Metric or method to increment** field, select from one of your previously created methods or metrics.
6. The **Increment by** field is pre-populated with **1**, however, change this to suit your own needs.
7. Click the **Create mapping rule** button.

#### Verification steps

- To verify your mapping rules, navigate to `[Your_product_name] > Integration > Methods & Metrics`. Each method and metric should have a check mark in the *Mapped* column.

## 19.5. ADDITIONAL RESOURCES

- For more details on API products and backends, see [Getting started with 3scale](#).

## CHAPTER 20. APPLICATION PLANS

Application Plans define the different sets of access rights you might want to allow for consumers of your API. These can determine anything from rate limits, which methods or resources are accessible and which features are enabled.

### 20.1. HOW TO CREATE AN APPLICATION PLAN

By default, when your 3scale account is created, you are given two plans: Basic and Unlimited. You can keep and edit these or create your own. You can create as many plans as you need.

To create a new application plan, follow these steps:

1. Navigate to **[Your\_product\_name] > Applications > Application Plans**
2. Click the **Create Application Plan** button.
3. On the **Create Application Plan** page, enter a name and a system name (system names must be unique) for your new plan.
4. If you want to require applications to get approval before they can access your API, select the **Applications require approval?** check box.

Once you have created a plan, you can provision [rate limits](#) and set up [paid plans](#).

### 20.2. SETTING UP A DEFAULT APPLICATION PLAN

After you have created all your plans, you can select a default plan for when people sign up to register their applications.

To select a default application plan:

1. Navigate to **[Your\_product\_name] > Applications > Application Plans**
2. From the drop-down list under *Default Plan*, select a plan.

If you do not indicate a default application plan, when a new user signs up to get access to your API, they will not be created an application by default. In other words, the user will not be able to get access to your API.

## CHAPTER 21. PROVISIONING PAID PLANS

One of the most popular ways to monetize an API, either products or backends, is by defining subscription fees based on usage. This section focuses on how to use application plans to provision pricing tiers, and how to set up a paid plan. It is also possible to apply pricing rules at the account, as well as at the product and backend levels – these topics are covered in advanced guides.

### 21.1. DECIDING YOUR PRICING MODEL

The first decision to make is how to differentiate between the tiers in your pricing model. The tiers can be driven by volume or usage, API functionality, access to other resources, or a combination of them:

- **Volume / Usage.** The most common way to differentiate between tiers is based on volume because volume usually has a strong correlation to value to the customer as well as cost to serve. You can apply a global hit count for calls on the product or a more granular measurement at the method level. *Volume drivers* are applied at the level of the global hits metric, or for individual methods under hits. Multiple pricing rules can be applied to any metric. Note that the hits calculation is cumulated over a one-month billing cycle.
- **Functionality.** You can enable or disable access to parts of your product depending on the tier. This is a good approach to distinguish between standard and premium levels.
- **Resources.** You can also create tiers based on access to any other resources that provide value to the customer or drive costs in your infrastructure – for example, gigabytes of bandwidth consumed, number of users, or transaction values. *Resource drivers* are similar to volume drivers but are applied on custom metrics.

Once you have decided on your pricing drivers, you must decide whether the tiers will be based on a flat rate subscription, a variable rate, or a one-off upfront charge. All three of the pricing drivers above are compatible with the one off, or monthly flat rate subscriptions. If you decide your pricing will be based on volume of hits or resource consumption, there will of course be a variable element to your pricing.

### 21.2. CONFIGURING AN APPLICATION PLAN WITH YOUR PRICING RULES

You can either create a new application plan or edit an existing one. When creating a new application plan, you can enter any upfront charges or flat rate subscriptions.

The screenshot shows the 'Create Application Plan' form in the Red Hat 3scale API Management Admin Portal. The form is titled 'Create Application Plan' and is located in the 'Applications' section of the 'Product: Echo API'.

The form fields are:

- Name:** A text input field.
- System name:** A text input field with a note: 'Only ASCII letters, numbers, dashes and underscores are allowed.'
- Applications require approval?:** A checkbox with the text: 'Set whether or not applications can be created on demand or if approval is required from you before they are activated.'
- Trial Period (days):** A text input field.
- Setup fee:** A text input field with a value of '0.00' and a 'USD' currency selector.
- Cost per month:** A text input field with a value of '0.00' and a 'USD' currency selector.

A 'Create Application Plan' button is located at the bottom right of the form.

In the edit application view, you can enter or modify the upfront charges and subscriptions.

Next, set up the pricing drivers you selected in [Section 21.1, "Deciding your pricing model"](#) :

1. Navigate to **[Your\_product\_name] > Overview > Applications > Application Plans**
2. Click on an application plan name.
3. Go to **Metrics, Methods, Limits & Pricing Rules** Here, pricing can be defined at the level of total hits, the granularity of methods, or for any custom metric.

If some of them already exist as metrics, you can edit the item.

Once you are finished setting up your pricing rules, click **Update Application plan**.

## 21.3. CREATING FURTHER PRICING TIERS

You can define an API paid plan with a single application plan. Usually this would be the case if all your pricing rules are defined by volume or resource drivers. However if you want to offer separate plans for different segments of your developer community, add more application plans.

The easiest way to do this is to copy the first application plan from the application plan overview page. This way, it will be pre-populated with all the existing metrics and pricing rules. The more care you take to create a full plan the first time, the more time you will save with the plan copy feature.

## 21.4. PROVISIONING THE PAID PLANS

In order to provision the plans, your developers must create new applications and select one of the new paid plans. You can also do this on their behalf from the admin console. For any existing applications, it is also possible to change from an existing plan to one of the new paid plans.

## 21.5. ADDITIONAL REFERENCES

In conjunction with flat-rate pricing plans it is common to differentiate between tiers using rate limits. This is explained in [provision rate limits](#)

## CHAPTER 22. PROVISIONING RATE LIMITS

Rate limits allow you to throttle access to your API resources, products and backends. You can configure different limits for separate developer segments through the use of application plans.

Once you have rate limits in place, these limits will control the responses a developer receives when they make authorization request calls to the 3scale back end.

### 22.1. CONFIGURING THE APPLICATION PLAN

If you do not have an application plan defined yet, create one first. Otherwise, select the plan you want to set rate limits for and click *edit*.

For more details about creating application plans, see [Application plans](#).

### 22.2. SETTING THE RATE LIMITS

To set the rate limits:

1. Navigate to **[Your\_product\_name] > Overview > Applications > Application plan**
2. Click on the name of the application plan you want to configure.
3. Scroll down to **Metrics, Methods, Limits and Pricing Rules**
4. Click on *Limits*.
5. Configure the limits on the product or the backend level.
6. When you are finished setting the limits you require, save your changes by clicking *Update Application plan*.

### 22.3. PUTTING THE NEW RATE LIMITS INTO ACTION

Now that you have your rate limits defined, the following will happen:

- If you have alerts configured, the new limits will be used to decide when notifications are sent.
- When you exceed the number of calls to the 3scale back end, the limits are considered and you will see the relevant error message. For more details about APIcast error messages, see [Configuring error messages](#).

Once your rate limits are operational, you will see the users who are reaching the limits on your dashboard, making it quick and easy to check for potential plan upgrade candidates. For more information about soft and hard limits, refer to the **Getting Started** guide in [Configure your API access policies with application plans](#) for the Advanced path.

### 22.4. MORE INFORMATION

Besides setting rate limits, you can also set variable pricing rules for the same metrics – see [provision paid plans](#)

## PART IV. BILLING

## CHAPTER 23. CONFIGURING BILLING SETTINGS

This document describes how the Billing feature works in Red Hat 3scale API Management.

Billing settings are divided into **Charging & Gateway** and **Credit Card Policies**, both can be found in **Audience > Billing** in the Admin Portal.

### 23.1. BILLING MODES (CHARGING & GATEWAY)

Billing in 3scale is based on the calendar month, and can be **Prepaid** and **Postpaid**.

- In the Prepaid mode, all fixed fees and setup fees are billed at the beginning of the month or at the beginning of any prorated billing period. Variable costs are always calculated and billed at the beginning of the following month.
- In Postpaid mode, all fixed fees as well as variable costs are billed at the beginning of the following month.

For more details see [Automated billing process](#).

### 23.2. CHARGING ENABLED (CHARGING & GATEWAY)

This setting 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 payment transaction will take place.

### 23.3. CURRENCY (CHARGING & GATEWAY)

Select the currency in which billing and the credit card transactions will be made. Please make sure that the selected currency is supported by your payment gateway. This setting is global and applies to all invoices and transactions; it is not possible to set different currencies for different developer accounts.

### 23.4. INVOICE FOOTNOTE (CHARGING & GATEWAY)

The text introduced in the Invoice footnote field will appear at the bottom of every PDF invoice. You can use this field to provide additional information about the charging and billing policy for your customers.

If the text of the footnote is changed, it doesn't automatically apply to the invoices, for which the PDF has already been generated. However, it is possible to apply the change by regenerating the PDF of the invoices.

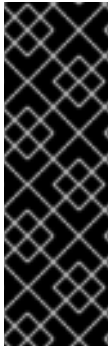
### 23.5. TEXT TO SHOW IF VAT/SALES TAX IS 0% (CHARGING & GATEWAY)

This field is used to add a text message to the PDF invoice, in case VAT / Sales Tax is 0% for the billed account. This line will only appear if the VAT / Sales Tax is set to 0 explicitly, otherwise it will not be shown. This text will also appear on the Invoice page in the Admin Portal.

See more about this setting in [VAT / Sales Tax section](#)

### 23.6. YAML CONFIGURATION FOR CURRENCIES

The **currencies.yml** file allows you to configure a list of currencies for your 3scale deployment. 3scale uses three-letter currency codes based on ISO 4217.



## IMPORTANT

- Ensure that the payment gateway supports the selected currency.
- 3scale integrates with the following payment gateways for credit card transactions:
  - Braintree
  - Stripe

### 23.6.1. Changing the currencies configuration in OpenShift

To change the currencies config, do the following:

#### Procedure

1. Add the source for the new content of **currencies.yml** as an entry of the **system** config map. The following example shows you how to add an additional currency, the *ARS - Argentine Peso* to the default list of currencies:

```
oc patch configmap system --type merge -p '{"data": {"currencies.yml": "\nproduction:\n'USD - American Dollar': 'USD'\n 'EUR - Euro': 'EUR'\n 'GBP - British Pound': 'GBP'\n 'NZD - New Zealand dollar': 'NZD'\n 'CNY - Chinese Yuan Renminbi': 'CNY'\n 'CAD - Canadian Dollar': 'CAD'\n 'AUD - Australian Dollar': 'AUD'\n 'JPY - Japanese Yen': 'JPY'\n 'CHF - Swiss Franc': 'CHF'\n 'SAR - Saudi Riyal': 'SAR'\n 'ARS - Argentine peso': 'ARS'\n\"}}"
```



## NOTE

To see an example of content for the **currencies.yml** config file, access the default YAML file: [currencies.yml](#). The file shows a default configuration of a new 3scale deployment:

```
base: &default
'USD - American Dollar': 'USD'
'EUR - Euro': 'EUR'
'GBP - British Pound': 'GBP'
'NZD - New Zealand dollar': 'NZD'
'CNY - Chinese Yuan Renminbi': 'CNY'
'CAD - Canadian Dollar': 'CAD'
'AUD - Australian Dollar': 'AUD'
'JPY - Japanese Yen': 'JPY'
'CHF - Swiss Franc': 'CHF'
'SAR - Saudi Riyal': 'SAR'
production:
<<: *default
preview:
<<: *default
```



2. Include the new **ConfigMap** entry **currencies.yml** in the **system-config** volume of **system-(app|sidekiq) DeploymentConfig**. This will mount the new the content inside the relevant containers and activate the new config.

```
export PATCH_SYSTEM_VOLUMES='{ "spec":{ "template":{ "spec":{ "volumes":{ "configMap":{
  "items":{ "key": "zync.yml", "path": "zync.yml",
  "key": "rolling_updates.yml", "path": "rolling_updates.yml",
  "key": "service_discovery.yml", "path": "service_discovery.yml",
  "key": "currencies.yml", "path": "currencies.yml"}}, "name": "system"}, "name": "system-
config"}}}}}}'
```

```
oc patch dc system-app -p $PATCH_SYSTEM_VOLUMES
oc patch dc system-sidekiq -p $PATCH_SYSTEM_VOLUMES
```

```
unset PATCH_SYSTEM_VOLUMES
```

### 23.6.2. Verifying the new currencies

To verify the currencies are available in your 3scale Admin Portal, do the following:

#### Procedure

1. Go to **Audience > Billing > Charging & Gateway**
2. Check that the list of currencies is available from the *Currency* drop-down list.
3. Choose the currency you intend to use.

## 23.7. BILLING PERIODS FOR INVOICE IDS (CHARGING & GATEWAY)

Invoices in 3scale have two types of identifiers:

#### Actual ID

which uniquely identifies the invoice in the database. This is a numeric ID, which appears in the invoice URL (i.e. [https://<dashboard\\_domain>/buyers/accounts/<account\\_id>/invoices/<invoice\\_id>](https://<dashboard_domain>/buyers/accounts/<account_id>/invoices/<invoice_id>)) and is used as the invoice ID in the Billing API.

#### Friendly ID

which appears on the invoice, is a user-friendly identifier. It should be unique per 3scale account, although it is technically possible to have duplicated friendly identifiers. In case that 3scale recognizes duplicate identifiers, you will see a warning message similar to: *This invoice id is already in use and should probably be changed*. If the friendly identifier is displayed as **fix** for more than 24 hours, contact support.

This setting defines the format of the Friendly ID of the invoice:

#### monthly (default)

**YYYY-MM-XXXXXXXX**, i.e. the ID includes the year, the month and the number of the invoice.  
Example: 2018-02-00000013

#### yearly

**YYYY-XXXXXXXX**, i.e. it only includes the year and the number. Example: 2018-00000001

The only effect of changing the Billing periods for invoice identifiers from *monthly* to *yearly* is changing

the friendly identifier format. This modification does not change the billing cycle period. Only monthly billing is supported by 3scale API Management. Changing the format of the invoice identifiers to yearly can be used, if there is such a requirement from your accounting department.

If you need to charge the customers yearly, billing should be handled manually – you can create a new invoice and add a line item with the yearly cost. If you prefer to use yearly charging, you may also want to set your application plans to be free, to make sure the invoices are not generated and/or charged automatically each month.

## 23.8. CREDIT CARD POLICIES

Here you can configure the path to the following pages:

- Legal Terms page
- Privacy page
- Refund page

## CHAPTER 24. CREDIT CARD GATEWAYS FOR PAYMENTS

As a 3scale API provider, define the payment gateways for credit cards to monetize subscriptions to your APIs.

### 24.1. CREDIT CARD GATEWAYS SUPPORTED BY 3SCALE

3scale integrates with the following payment gateways for credit card transactions:

- Braintree
- Stripe

Online payment services must display a mandate when collecting credit card data. A mandate is a record of the permission a customer has given you to debit their payment method. The mandate should state clearly that the supplied payment method will be used to collect subsequent payments for services. For more information regarding mandates for Stripe and Braintree, see the external documentation links for the respective payment gateways listed under [Additional resources](#).

#### Additional resources

- Stripe: [Accept a SEPA Direct Debit payment](#)
- Braintree: [Recurring Transactions](#)

### 24.2. CONFIGURING STRIPE AS A CREDIT CARD GATEWAY

As a 3scale API provider, configure the Admin Portal and the Developer Portal with Stripe as a payment gateway to receive payments from subscriptions to your APIs by using Stripe as a credit card gateway.

#### Prerequisites

- You must have a Stripe account.
  - Stripe recommends you use separate Stripe sub-accounts for each business or project.
  - See the Stripe documentation for [Multiple Accounts](#).
- You must have Stripe administrator permissions.

#### Procedure

To configure 3scale with Stripe as a payment gateway, follow these steps:

1. [Generating an access token with the Billing API scope in the 3scale Admin Portal](#) .
2. [Obtaining the keys and the webhook secret from Stripe](#) .
3. [Configuring charging in the 3scale Admin Portal](#) .
4. [Editing credit card details in the 3scale Developer Portal](#)
5. [Updating the text of \*\*unsuccessfully charged\*\* email response](#).

## 24.2.1. Generating an access token with the Billing API scope in the 3scale Admin Portal

1. In the 3scale Admin Portal, go to **Account Settings > Personal > Tokens**
2. Create a *Read & Write* token with the Billing API scope:
  - a. Click **Add Access Token**.
  - b. Specify a name for the token.
  - c. Choose the scope: *Billing API*.
  - d. Select the permission level: *Read & Write*.
  - e. Click **Create Access token**.
  - f. Copy the access token.
    - Ensure you copy the access token to a file text. The access token will not be displayed afterwards.
  - g. To finish the token generation, click **I have copied the token**.

[Back to Procedure](#) .

## 24.2.2. Obtaining the keys and the webhook secret from Stripe



### NOTE

- Configuring the webhook in Stripe is required.
- Use the webhook to notify 3scale that the payment has succeeded.
- 3scale then updates the state of the invoice and prevents further attempts to charge.

In your Stripe account, get the *Secret Key*, and the *Publishable Key*:

1. Open the Stripe dashboard.
2. [Find your API keys by following the instructions in the Stripe documentation](#) .
3. Copy the *Secret Key*, and the *Publishable Key*

Still in your Stripe account, create a *Webhook Signing Secret*:

1. Go to **Developers > Webhooks**.
2. Click **Add endpoint**
3. Fill in with the following endpoint URL:

```
https://<Your-provider-admin-domain>/api/payment_callbacks/stripe_callbacks?  
access_token=<value-of-access-token>
```

4. In **Events to send**, add **payment\_intent.succeeded**.
5. Click **Add endpoint**.
6. Click to reveal the signing secret of the webhook you just created and take note of this secret. This is the *Webhook Signing Secret*.

[Back to Procedure](#).

### 24.2.3. Configuring charging in the 3scale Admin Portal

In the 3scale Admin Portal:

1. Go to **Audience > Billing > Charging & Gateway**
2. Select *Charging enabled* and click **Save**.
3. In **Credit card gateway > Gateway**, select *Stripe* as the gateway.
4. Add the *Secret Key*, the *Publishable Key* and the *Webhook Signing Secret* you obtained from your Stripe account in [Section 24.2.2, "Obtaining the keys and the webhook secret from Stripe"](#).
5. Click **Save**.

[Back to Procedure](#).

### 24.2.4. Editing credit card details in the 3scale Developer Portal

1. Login to the 3scale Developer Portal with a developer account.
2. Go to **Settings > Credit Card Details**.
3. Add the following credit card details: credit card number, expiration date and CVC.
4. Click **Save details**.

[Back to Procedure](#).

### 24.2.5. Updating the text of unsuccessfully charged email response

In connection with the fixes for SCA payments, the text of the **invoice\_messenger\_unsuccessfully\_charged\_for\_buyer.text.liquid** email requires a manual update in 3scale 2.10.

1. In the 3scale Admin Portal, go to **Audience > Messages > Email Templates**
2. Select *Invoice charge failure for buyer with retry*.
3. Click **Override**.
4. Update the template message: This is the complete text to be used in the unsuccessfully charged email response:

Dear {{ account.name }},  
  
Thank you for using our service.

We're sorry to inform you that your last payment was declined.  
This may have been caused by a few common reasons:

- A new authentication policy enforced by your bank
- An expired credit card
- Insufficient funds on the account

To continue using your service, verify the status of your credit card and update or re-enter the credit card details at `{{payment_url}}`.

If you need help, don't hesitate to contact us at `{{ provider.finance_support_email }}`.

Best regards,  
The `{{ provider.name }}` API Team

#### 5. Click **Create Email Template**

With these steps, you have updated the email template for the **unsuccessfully charged** email response.

[Back to Procedure](#) .

#### Additional resources

- [Stripe Docs: API keys](#)

## 24.3. CONFIGURING BRAINTREE AS A CREDIT CARD GATEWAY

As a 3scale API provider, configure the Admin Portal and the Developer Portal with Braintree as a payment gateway to receive payments from subscriptions to your APIs by using Braintree.

#### Prerequisites

- You must have an account with [Braintree](#).
- If you want to ensure a secure checkout for your customers with 3D Secure (3DS), you must have 3DS enabled in your Braintree account before you enable 3DS for 3scale.
  - By default, both 3scale and Braintree have 3DS as **off**(disabled).

#### Procedure

To configure 3scale with Braintree as a payment gateway, follow these steps:

1. [Obtaining the keys and the merchant identifier from Braintree](#) .
2. [Configuring charging in the 3scale Admin Portal](#) .
3. [Editing credit card details in the 3scale Developer Portal](#)

#### Obtaining the keys and the merchant identifier from Braintree

From your Braintree account, get the **Public Key**, the **Merchant ID**, and the **Private Key**. For more information about obtaining these values, refer to the Braintree documentation listed under [Additional resources](#).

[Back to Procedure](#) .

## Configuring charging in the 3scale Admin Portal

In the 3scale Admin Portal:

1. Go to **Audience > Billing > Charging & Gateway**
2. Select *Charging enabled*.
3. Select the currency.
  - The currency type specified on the 3scale Billing page must match the currency type used in your Braintree merchant account.
4. Click **Save**.
5. Under **Credit card gateway > Gateway**, select *Braintree* as the gateway.
6. Add the **Public Key**, the **Merchant ID** and the **Private Key** you obtained from your Braintree account in [Obtaining the keys and the merchant identifier from Braintree](#) .
7. To enable 3DS, select **3D Secure Enabled**.
8. Click **Save changes**.

[Back to Procedure](#) .

## Editing credit card details in the 3scale Developer Portal

As a 3scale API consumer, add or edit the credit card details in the 3scale Developer Portal. To match the financial details with the entity that has issued your credit card, all the fields listed in this window are mandatory.

1. Login to the 3scale Developer Portal with a developer account.
2. Go to **Settings > Credit Card Details**
3. Click the **Add Credit Card Details and Billing Address** link.
4. Add the payment details: first name, last name, phone.
5. Add the credit card details: credit card number, expiration date and CVC.
6. Add the billing address details: company, street address, zip/postal code, city and state/region. Then, select the country.
7. Click **Save details**.
8. If you are prompted, complete two-factor authentication (2FA) for your purchase. For example, if your bank has enabled the SMS 2FA option, you will have to complete the authentication process using this method.

[Back to Procedure](#) .

## Additional resources

- For more details about obtaining the **Public Key**, the **Merchant ID** and the **Private Key** from your Braintree account, see [Braintree Articles: Important Gateway Credentials](#).

## 24.4. ALLOWING PAYMENTS OF REJECTED INVOICES VIA THE DEVELOPER PORTAL

As a 3scale API provider, allow the payment of rejected invoices via the Developer Portal. To enable these payments, update the **Invoices** template in the Admin Portal. Note that this procedure is intended for existing instances of the Developer Portal.

### Prerequisites

- You must have administrator permissions for 3scale.
- You must have an account with [Stripe](#) or [Braintree](#).

### Procedure

To allow payments of rejected invoices via the Developer Portal, follow these steps:

1. In the 3scale Admin Portal, go to **Audience > Developer Portal > Content**
2. Edit **Root > Invoices > Show template**
3. Replace these lines of code:

```
<a href="{{ urls.invoices }}">
  <i class="fa fa-chevron-left"></i>
  Cancel
</a>
{{ invoice.period_begin | date: '%B, %Y' }} Invoice
```

With this snippet:

```
<div class="clearfix">
  <a href="{{ urls.invoices }}">
    <i class="fa fa-chevron-left"></i>
    Cancel
  </a>
  {{ invoice.period_begin | date: '%B, %Y' }} Invoice
  {% if invoice.pay_now? %}
    <a href="{{invoice.url}}/payment" class="pull-right btn btn-success pay-invoice-btn">Pay
invoice</a>
  {% endif %}
</div>
```

## 24.5. TROUBLESHOOTING ISSUES WITH CREDIT CARD GATEWAYS

As a 3scale API provider working with either Stripe or Braintree as payment gateways, you can troubleshoot some issues with these credit card gateways.

### Stripe



- 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]**

### Braintree

- In case your Braintree account is in the sandbox mode and you encounter any problems, you will have to change it to production.
- For credit cards that are stored in the 3scale Developer Portal without 3D Secure (3Ds) enabled, these are the suggested solutions to integrate 3scale with Braintree:
  - a. 3scale API providers: follow the steps listed in [Configuring charging in the 3scale Admin Portal](#).
  - b. 3scale API consumers: follow the steps listed in [Editing credit card details in the 3scale Developer Portal](#).
- 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]**

## CHAPTER 25. PRICING

This section describes different ways you can charge your developers for using your API.

### Setup fee

is a one-time charge applied upon subscription to the service, it is not charged when switching to another plan. It appear in the invoice/credit card only on the first month of a subscription. Can be configured on application plans, service plans, and account plans.

### Cost per month

is a recurring cost charged monthly. It is prorated if the subscription occurred in the middle of the month. Sometimes it is referred to as "fixed fee". Can be configured on application plans, service plans, and account plans.

### Variable costs

are the costs derived from the pricing rules applied to each method/metric configured in the application plan. They are based on the usage of your API and therefore cannot be known in advance, only when the billing period has concluded. Only available on application plans.

### Example

If you have a plan with monthly cost of \$10, but want to charge your developer a \$5 setup fee. The initial charge would be for \$15, while all subsequent charges would be for \$10.

## 25.1. PRICING RULES

Pricing rules define the cost of each API request. Multiple pricing rules on the same metric divide up the ranges of when a pricing rule applies. Pricing rules are based on calendar month, and the counter is reset at 00:00 UTC of the 1st day of each month.

### Example 1

Until 100 calls per month (from 1 to 100) each API call can be charged at \$0.04, and starting from the call 101 (from 101 to infinity) the calls are charged at \$0.10.

### Example 2

The first 1000 calls are not charged (cost \$0), because they are included in the plan which has a monthly fixed cost. Starting from call 1001, each call is charged at \$0.50.

### Example 3

Calls from 1 to 100 are charged at \$0.30, 100 to 500 – at \$0.40, and 500 and further – at \$0.50.



### NOTE

Pricing rules are defined for metrics and methods. The actual API requests are mapped to these metrics and methods through the Mapping Rules.

## 25.2. SETTING PRICING RULES

1. Go to [Your\_API\_service] > Applications > Application Plans

2. Select an existing application plan or create a new one.
3. In the section **Metrics, Methods, Limits & Pricing Rules** click **Pricing (x)** to open the pricing section.
4. Click new pricing rule.
5. Set the values From, To and Cost per Unit and click **Create pricing rule**

Repeat the last two steps to create all the necessary pricing rules ranges.

Leave the To field empty to set the rule to "to infinity".

The maximum number of decimals is set to 4 for the cost of metric, if a number is added with more decimals, the value is rounded to a number with 4 decimal places.

### 25.3. UPDATE EXISTING PRICING RULES

1. Click on edit.
2. Make the necessary adjustments to the From, To and Cost per Unit fields.
3. Click Update pricing rule.

## CHAPTER 26. BILLING

The billing process creates invoices under the developer account, if it is subscribed to any paid services. The invoices are charged using the configured payment gateway.

### 26.1. LISTING INVOICES

In the **Audience > Billing > Earning by month** section you can find an overview of your due and received payments, calculated as the sum of all invoices issued that month, depending on their state:

#### Total

all invoices excluding cancelled

#### In process

Open, Finalized and Pending invoices

#### Overdue

Failed and Unpaid invoices

#### Paid

Paid invoices

By clicking on a specific month you can list all invoices for that month.

You can also see the list of invoices by going to **Audience > Billing > Invoices** where you can find invoices by their ID (friendly ID), the account name, and also filter by month and state.

The invoices of a specific developer account can also be found under **X Invoices** in the breadcrumbs of the account details page at **Audience > Accounts > Listing > [selected account]** where X is the number of invoices the account has.

### 26.2. INVOICE VIEW

If you click on the invoice friendly ID from the invoices list, you will see the details of the invoice.

#### 26.2.1. Invoice details

The title includes the month and year of the invoice, and also shows whether the invoice was created manually or by automated billing process, e.g. **Invoice for February 2018 (manually created)**

The **Details** block below shows the friendly ID of the invoice, invoice state, dates when the invoice was finalized and issued, and when it is due and paid on. If the PDF has already been generated, the link to download the PDF will be also shown.

The **Issued by** and **Issued to** blocks shows the name and the address of the API provider and the developer account accordingly.

Below a block with **line items** will be shown, including calculations of the VAT/Sales Tax, total cost, and if the text in "Text to show if VAT/Sales Tax is 0%" was configured, it will also be shown here.

In the **Transactions** block of the invoice you can see the list of all credit card transactions attempts, including the status of the transaction, its timestamp, reference number, message provided by the payment gateway and the amount. You can find the transactions in your payment gateway administrator console using the reference number.

## 26.2.2. Editing invoices

If the invoice is in Open or Finalized state, you can edit some details of the invoice.

By clicking on the Edit link at the top right corner of the invoice view, you can change the friendly ID and the billing period.

You can also add and delete invoice line items (except calculated costs – total, VAT/sales tax, total with VAT/sales tax). To add a new line item, click on Add, and fill in Name, Quantity (does not affect the cost), Description and Cost.

The links at the bottom of the invoice view can trigger actions on the invoice – different actions are available depending on the current state of the invoice, e.g. Generate PDF, Issue invoice Cancel invoice, Regenerate PDF, Charge, Mark as paid etc. These actions (except PDF-related) will change the state of the invoice.

## 26.3. INVOICE STATES

The invoice can be in one of the following states:

### Open

the invoice is created, but the automated calculations has not yet been finalized.

### Finalized

the invoice has all the current billing period charges added to it.

### Pending

the invoice has been issued to the developers and is awaiting payment.

### Unpaid

the invoice has been charged unsuccessfully, but pending automatic retry.

### Paid

the invoice has been charged successfully.

### Failed

the invoice has been charged unsuccessfully, and automatic retry will not be attempted anymore.

### Cancelled

has been cancelled by an admin.

## 26.4. AUTOMATED BILLING PROCESS

In 3scale, the billing process runs daily. It generates invoices, changing their states according to the billing flow, and performs charges using the configured payment gateway.

The billing flow is a bit different for Prepaid and Postpaid mode, and as billing in 3scale is based on calendar month, there are special events that happen on the first day of the month.

### 26.4.1. On the first day of each month

#### Postpaid

- Bill variable cost for the **previous** month: The cost is included as a line item to the open invoice.
- Finalize open invoices for the **previous** month

- Bill fixed cost for the **current** month: Create a new invoice for the current month in *Open* state.

### Prepaid

- Bill fixed cost (**current** month)
- Bill variable cost (**previous** month)

Notifications are sent to the API admins about the invoices finalized in the beginning of the month, so they can review the invoices and make necessary adjustments.

All actions that are performed every day also happen on the first day of the month in addition to the ones described above.

### 26.4.2. Every day

- Bill expired trials and new contracts that have not yet been billed: Invoices in *Open* state for the current month are created.
- **Prepaid** only: Finalize all open invoices: Status changes to *Finalized*.
- Issue invoices: Status changes to *Pending*.
  - The invoices are generally issued 2-3 days after they were finalized. The "*Issued On*" date of the invoice is set to current date, and the "*Due On*" date (when the invoice will be charged) is set to the *Issued On* + 2 days.
  - When the invoice is issued to the developers, they receive an email notification, and can see the issued invoice in the Developer Portal.
- Charge invoices
  - The invoices in *Unpaid* and *Pending* states are charged, if the *Due On* date is today or earlier.
  - If the payment failed, the invoice state is changed to *Unpaid*. Charging will be retried again in 3 days. After 3 unsuccessful retries, the invoice state is changed to *Failed*, and charging is not retried anymore.
- Notify about expired credit cards
  - The developer accounts whose credit cards are going to expire soon are sent email notifications.

### 26.4.3. Automatic and manual invoices

The invoices that are generated by the automated billing process have an (*automatically created*) label in the invoice header. For example: *Invoice for January 2019 (automatically created)* .

The invoices that are generated manually are marked with (*manually created*) on the invoice details page.

The automated billing process may use existing invoices in **open** state for the current month to create additional line items, but only for automatically created invoices. The invoices created manually will not be updated by the automated billing process.

### 26.4.4. Mid-month upgrades

If an application (or account/service subscription) is upgraded in the middle of the month, the monthly cost is prorated according to the number of days left in the month. The limits configured on the application plan are not prorated.

If the application is upgraded from a free to a paid plan, the next time billing runs a new invoice will be generated including the prorated monthly cost.

When the application is upgraded from a paid plan to a more expensive paid plan, the behavior depends on several factors:

- The mode of Billing: **Prepaid** or **Postpaid**
- When the plan change is made

#### 26.4.4.1. Prepaid billing

1. If the application plan is changed **on the same billing day (a billing day starts at 8am UTC)** as it was created, and **so it has never been invoiced before** the fixed costs for the old plan are included to the invoice, and are discounted with a 'Refund' line item. The fixed costs for the old plan are also added to the invoice.

**Example:** A customer signed up for the Plan A (200\$) on the first day of the month, and upgraded to the Plan B (300\$) *on the same day*. One invoice will be generated in this case, and it will contain the following line items:

Description	Cost
Fixed fee ('Plan A')	200
Refund ('Plan A')	-200
Application upgrade ('Plan A' to 'Plan B')	300
Total	300

Note that if the customer signed up on another day of the month, the cost of and refund of 200 would be prorated.

2. If the application plan is changed **after the invoice was already issued** for this application:
  - In case of an **upgrade**, the developer will be issued two invoices: one for the initial charge, and another one for the upgrade.

**Example:** A customer signed up for the Plan A (200\$) on the first day of the month, and then upgraded to the Plan B (300\$) in the middle of the month. The following invoices will be generated:

Description	Cost
Fixed fee ('Plan A')	200

Description	Cost
Total	200

Description	Cost
Refund ('Plan A')	-100
Application upgrade ('Plan A' to 'Plan B')	150
Total	50

In the second invoice, the refunded cost (100\$) and the new cost (150\$) are prorated, as the upgrade is made in the middle of the billing period.

- Refunding on application **downgrade** (change to a plan with a lower cost) is not supported at the moment.

#### 26.4.4.2. Postpaid billing

In Postpaid billing mode a single invoice will be issued, and it will include the *Refund* and *Application upgrade* line items.

IMPORTANT: This behavior was introduced on **April 20th, 2018** with the following changes:

- A bug was fixed where the initial charge (for the initial application plan) was not included in the invoice when the application was upgraded on the same day as created.
- Previously only one line item was added on application upgrade, including the difference between the cost of the new and the old plan. For example, in the scenario 2 described in **Prepaid billing** section above (application upgrades from Plan A – 200\$, to Plan B – 300\$), the second generated invoice would be:

Description	Cost
Application upgrade ('Plan A' to 'Plan B')	50
Total	50

where 50\$ is the difference between the prorated costs of the new and old plan for the rest of the month (150\$ - 100\$).

After **April 20th, 2018**, the calculations are reflected more clearly in the invoice (including refund and charge separately), while the total cost remains the same as before.

## 26.5. ENABLE/DISABLE BILLING/CHARGING PER ACCOUNT

Automated billing process generates invoices for all developer accounts subscribed to paid services.



Charging can be enabled or disabled globally. See [Configuring Billing Settings](#) ), however it is also possible to enable or disable both billing and charging per developer account. To do that, navigate to the account page and click the corresponding button (Enable / Disable) against:

- *Monthly billing is enabled/disabled*
- *Monthly charging is enabled/disabled*

If charging is disabled, and billing is enabled, the developer will be issued invoices, but will not be charged. This is useful in case you handle charging separately (e.g. with wire transfer).

If both are disabled, the developer will not be issued or charged.

If billing is disabled, and charging is enabled, the developer will not be issued any new invoices, but all the outstanding ones (*Pending* and *Failed*) will still be charged.

Here in the **Billing Status** block you can also see whether the credit card details are configured on the account. In case credit card details are configured, the month and the year of card expiration is shown. The presence of the credit card details may affect the Developer Portal flow on sign up and plan change (see more in [Section 26.9, "Credit card flow"](#) ).

## 26.6. PLANS WITH A TRIAL PERIOD

A trial period (in days) can be set for paid plans to delay the charging for the specified amount of days.

You can see the trial period status for the application in the application listing (**[Your\_product\_name] > Applications > Listing**) in the *State* column as *"live - trial expires in N days"* , and in the application details page in the *Trial days left* field.

During the trial period the application can use all the features and usage limits defined by the plan without restrictions.

Once the trial period is set, it is not possible to cancel or extend it. Upgrading to another plan will not reset the trial days counter.

10 days before the trial period expires, an email notification is sent to the developer account notifying about the upcoming trial expiration. The template of this email can be configured in **Audience > Messages > Email Templates**: *"Application trial period expired for buyer"* for the application plans, and *"Service trial period expired for buyer"* for service plans with trial period. The developer should either remain on the current plan, or switch to another plan, or contact the API provider if they decide to cancel the subscription.

Once the trial period is over, the next time automated billing runs (see [Section 26.4, "Automated billing process"](#)) the developer will be charged according to the costs set for the plan that the application or service are currently on. Note that the application will **not** be blocked or suspended after the trial period expires, and will continue to work.

While a trial period technically can be set for free plans, it is not recommended as the feature loses its usefulness.

## 26.7. VAT RATE/SALES TAX

Value-Added Tax (VAT) is a tax that is applied on goods and services sold within European Union. In other countries there is a similar tax, for example, Sales tax. API providers who charge their customers for using their API service often need to charge the tax as well and reflect it in the invoices to comply with the existing regulations.

3scale billing can automatically include VAT in the invoices, when configured.

### 26.7.1. Configure VAT rate field

VAT rate is the number (in percentage) that will be applied to the total cost in the invoice. To configure VAT rate, follow these steps in the Admin Portal, from the Dashboard:

1. Go to **Audience > Accounts > Fields Definitions**
2. Click **Create** in the *Account* section.
3. Select **vat\_rate** in the [new field] drop-down list.
4. Set a value for **Label**.
5. Configure the checkboxes to indicate whether the field should be *Required*, *Read only* or *Hidden* for the developers.  
**NOTE:** the "Choices" field only accepts strings, so this field cannot be used.
6. Click **Create**.

### 26.7.2. Configure VAT code field

VAT code is the VAT identification number. To configure the VAT code follow these steps in the Admin Portal, from the Dashboard:

1. Go to **Audience > Accounts > Fields Definitions**
2. Click **Create** in the *Account* section.
3. Select **vat\_code** in the [new field] drop-down list.
4. Set a value for **Label**.
5. Configure the checkboxes to indicate whether the field should be *Required*, *Read only* or *Hidden* for the developers.
6. Click **Create**.

### 26.7.3. Set VAT values for an account

Once VAT code and VAT rate fields are added, you can set the corresponding values under the developer account Edit form. Depending on the selected checkboxes in the fields definitions the fields can also be visible and/or editable by the developers in the Developer Portal.

VAT code can be any string.

VAT rate must be a number. Either integer or decimal numbers are accepted (e.g.: 21, 23.5 etc.) and they represent the percentage of the cost that will be included as VAT.

### 26.7.4. Invoices with VAT

When an invoice is generated for a developer account that has a non-zero VAT rate configured, the invoices will include the following line items:

```

_ Total cost (without <VAT rate>)_ +
_<VAT rate> Amount_ +
_ Total cost (<VAT rate> <VAT rate value>% included)_

```

<VAT rate> will be replaced with the Label set for the VAT rate field, and <VAT rate value> is the value configured for the account the invoice is issued for.

The VAT code will be included in the PDF version of the invoice with the label, specified in as the Label in the VAT code field definition. The value will correspond to the value set in the developer account details.

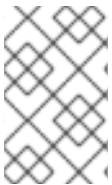
The fields **vat\_rate** and **vat\_code** can also be read and written by the 3scale Account Management API.

## 26.8. DEVELOPER PORTAL VIEW

The developers can manage their Credit Card information and see their invoices in the Developer Portal.

The Finance switch in **Audience > Developer Portal > Feature Visibility** has to be in *Visible* state for the developers to see the billing-related elements in the Developer Portal.

A logged-in user can update the credit card details at **Settings > Credit Card Details**. Depending on the payment gateway configured the form may be different, and the user may need to enter the billing address first.



### NOTE

Credit card details are not stored in the 3scale database. These details are managed by the payment gateway. Only the last 4 digits of the credit card number, the expiration date and the reference provided by the payment gateway are stored in the 3scale database.

A developer can only have one credit card card on the Developer Portal.

Credit card details cannot be deleted from the Developer Portal. The user of the developer account who wants to delete his credit card details would need to request the API provider to delete the credit card details from the system.

To see the invoices, the user can go to **Settings > Invoices**, and from there they can either show the details for each invoice, or download the invoice PDF.

## 26.9. CREDIT CARD FLOW

### 26.9.1. Sign up for a paid plan

When a developer signs up for a paid plan, they are requested to enter the credit card details before they are allowed to see the application credentials. Once the developer logs in to the Developer Portal for the first time, they will be redirected to the Credit Card Details page. Trying to access any other developer account page will result in a redirection to the Credit Card Details page again.

You can use Liquid tags to hide all of the menu items, except the **Credit Card Details** tab, by customizing the corresponding Developer Portal templates.

The **current\_account** Liquid drop exposes the method **requires\_credit\_card\_now?** which returns **true** if the credit card details are missing, but are required (only if Billing is configured in the Admin Portal), and **false** otherwise.

You can hide any menu items and other user interface elements in the **submenu** and **users\_menu** partials by wrapping them with the following condition:

```
{% unless current_account.requires_credit_card_now? %}
...
{% endunless %}
```

### 26.9.2. Upgrade from a free to a paid plan

For plan changes of the existing applications there are multiple options that can be configured, including changing the plan directly by the developer or requesting a plan change. In case the application is upgraded from a free to a paid plan, it is important to ensure that the developer enters the credit card details before being able to upgrade. This can be configured in **[Your\_product\_name] > Integration > Settings, Application Plan Changing** section by the following setting:

*Change plan directly if developer has credit card, otherwise:*

- Only request a plan change
- Request Credit Card entry for paid plans

Select the first option if you want to only allow the developer to request the change, and perform the upgrade manually after the credit card details are entered.

Select the second option if you want the developer to be notified that credit card details need to be entered before upgrading to a paid plan. In the plan selector widget the developer will see a message "You cannot change plan until you enter your Credit Card details" pointing to the credit card details form.

## 26.10. "BILLING ADDRESS" FIELD

When the developer enters the billing address on the Credit Card Details in the Developer Portal, this address is stored separately from the "legal" account address.

By default the legal address appears in the Invoices in the *Issued to* field. However, in case the developer does not have the legal address configured, but only the billing address, the latter will be used on the invoices. The organization name is considered part of the address in this case.

By default the billing address is not visible in the Admin Portal or through the product API; however you can add it as follows:

1. Go to **Audience > Accounts > Fields Definitions**
2. Under the *Account* block click **Create**.
3. Select 'billing\_address' from the drop-down list, add the label, check the Read only checkbox and click **Create**.

Now the **<billing\_address>** field appears in the XML model for Accounts in the Account Management API and Webhooks. The corresponding billing address will be visible as read-only in the Account and the Account edit pages of the Admin Portal.

Billing address can be changed by the developer in the Developer Portal, in the Credit Card Details section. The admin cannot change the billing address in the Admin Portal, but it is possible to do so with the *Account Edit* endpoint of the Account Management API using the following fields (as "additional fields"):

billing\_address\_name  
billing\_address\_address1  
billing\_address\_address2  
billing\_address\_country  
billing\_address\_city  
billing\_address\_state  
billing\_address\_zip  
billing\_address\_phone

## CHAPTER 27. EMAIL NOTIFICATIONS

Different events related to billing trigger email notifications for API providers and developers.

### 27.1. PROVIDER NOTIFICATIONS

The users of the 3scale account (admins and members with *Billing* permission) can subscribe to or unsubscribe from the notifications related to billing at **Account Settings (gear icon on the top-right) > Personal > Notification Preferences**, under Billing section:

#### **Action required: review invoices**

Sent a few days before end of billing cycle so you can review invoices before they are being sent to customers.

#### **Customer downgraded**

Sent when a customer changes to a plan with a lower monthly fixed price.

#### **Expiring credit card**

Sent when a customer's credit card is about to expire.

#### **Payment error (retry)**

Sent when payment fails, resulting in an unpaid invoice and a retry.

#### **Payment error (final)**

Sent when the final retry of a payment fails, resulting in a failed invoice.

Note: All admin users of the 3scale account will receive notifications regarding billing, if they are subscribed to them.

### 27.2. DEVELOPER EMAILS

The email notifications sent to the developer accounts can be configured at **Audience > Messages > Email Templates**. The following emails are available:

#### **Credit card expired notification for buyer**

Sent when the credit card is due to expire soon.

#### **Invoice charged successfully for buyer**

Sent when the invoice has been successfully charged.

#### **Invoice charge failure for buyer with retry**

Sent when the invoice charge has failed, and the invoice is in Failed state, which means that the charging will be retried again.

#### **Invoice charge failure for buyer without retry**

Invoice charge has failed for the 3rd time, the invoice has passed to Unpaid state and will not be retried again.

#### **Upcoming invoice charge for buyer**

Sent when the invoice is issued for the developer.

All admin users of the developer account will receive the above notifications.

#### 27.2.1. Billing email address

You can configure an email address that your customers can contact for resolving any issues with billing (e.g. [billing@example.com](mailto:billing@example.com)), in **Audience > Messages > Support Emails** in the *Finance support email* field.

The email templates reference the email address with the Liquid drop **{{ provider.finance\_support\_email }}**.

## CHAPTER 28. BILLING API

The Billing API provides a way to automate common billing processes.

All the endpoints of the Billing API can be found in the Admin Portal under **Documentation (?) > 3scale API Docs > Billing API**.

The Billing API requires a valid access token which meets the following requirements:

- it should belong to either an admin user of the provider account, or a member user with "Billing" permissions
- it should include "Billing API" scope

Note that when an invoice ID is required as a parameter, it refers to the invoice ID, and **not** the Friendly invoice ID.

The XML response of the API endpoints is mostly self-explanatory, and the fields of the Invoice represent the same information as in the web and PDF representation.

Some notable fields of the response:

- **creation\_type**: can have the following values: *'manual'* for invoices created manually or *'background'* for invoices created by the 3scale automated billing process
- **provider**: the details of the API provider (the admin account), corresponds to the *Issued by* section of the invoice.
- **buyer**: the details of the developer account, corresponds to the *Issued to* section of the invoice.

The XML representation of the invoice also includes the list of Line Items under the **line-items** field.

For some line items (typically the ones created automatically) apart from the expected name, description, quantity and cost (price), you can see the following:

- **type**: the type of the line item, can have the following values:
  - **LinItem::PlanCost** – for line items for fixed plan costs
  - **LinItem::VariableCost** – for line items for variable costs
- **metric\_id**: for variable costs line items – the ID of the metric that the cost is associated with
- **contract\_id**: the ID of the service or application that the cost is associated with



## PART V. SERVICE DISCOVERY: FROM OPENSIFT TO 3SCALE

Service Discovery is a 3scale feature that helps you import services from an OpenShift cluster. The private endpoint of the OpenShift service becomes an API backend and a new API product bundling this API backend. Additionally, the specification of the service which is based on OpenAPI Specification (OAS), is imported as an ActiveDoc of the API product.

## CHAPTER 29. SERVICE DISCOVERY

With the Service Discovery feature provided by 3scale, you can import API services from OpenShift.

### 29.1. ABOUT SERVICE DISCOVERY

When Service Discovery is configured, 3scale scans for discoverable API services that are running in the same OpenShift cluster and automatically imports the associated API definitions into 3scale. Additionally, 3scale can update the API integration and its specification, based on OpenAPI Specification (OAS), to resynchronize them with the cluster.

Service Discovery offers the following features:

- Uses the cluster API to query for services that are properly annotated for discovery.
- Configures 3scale to access the service using an internal endpoint inside the cluster.
- Imports the API service specification as a 3scale ActiveDocs.
- Supports OpenShift and Red Hat Single Sign-On (RH SSO) authorization flows.
- Works with Red Hat Fuse, starting with Fuse version 7.2.

When you import a discoverable service, it keeps its namespace within the project it belongs to. The imported service becomes a new customer-facing API, product, and its corresponding internal API, backend.

- For 3scale on premises, the 3scale API provider may have its own namespace and services. Discovered services can co-exist with 3scale existing and native services.
- Fuse discoverable services are deployed to the Fuse production namespace.

### 29.2. CRITERIA FOR A DISCOVERABLE SERVICE

If you want to have 3scale discover an API service in an OpenShift (OCP) cluster, said OCP service must meet the criteria for each element below:

#### Content-Type header

The API specification's **Content-Type** header must be one of the following values:

- **application/swagger+json**
- **application/vnd.oai.openapi+json**
- **application/json**

#### OpenShift Service Object YAML definition

- The OpenShift Service Object YAML definition must include the following metadata:
  - The **discovery.3scale.net** label: (required) Set to "true". 3scale uses this label when it executes the selector definition to find all services that need discovery.
  - The following annotations:

**discovery.3scale.net/discovery-version:** (optional) The version of the 3scale discovery process.

**discovery.3scale.net/scheme:** (required) The scheme part of the URL where the service is hosted. Possible values are "http" or "https".

**discovery.3scale.net/port:** (required) The port number of the service within the cluster.

**discovery.3scale.net/path:** (optional) The relative base path of the URL where the service is hosted. You can omit this annotation when the path is at root, "/".

**discovery.3scale.net/description-path:** The path to the OpenAPI service description document for the service.

For example:

```

metadata:
  annotations:
    discovery.3scale.net/scheme: "https"
    discovery.3scale.net/port: '8081'
    discovery.3scale.net/path: "/api"
    discovery.3scale.net/description-path: "/api/openapi/json"
  labels:
    discovery.3scale.net: "true"
name: i-task-api
namespace: fuse

```

- If you are an OpenShift user with administration privileges, you can view the API service's YAML file in the OpenShift Console:
  1. Select **Applications > Services**.
  2. Select the service, for example **i-task-api**, to open its Details page.
  3. Select **Actions > Edit YAML** to open the YAML file.
  4. When you have finished viewing it, select **Cancel**.

### Clusters with the **ovs-networkpolicy** plugin

- To allow traffic between the OpenShift and 3scale projects, clusters that have the **ovs-networkpolicy** plugin require **NetworkPolicy** objects created within their application project.
- For more information about configuring a **NetworkPolicy** object, see [About network policy](#)

## 29.3. CONSIDERATIONS FOR CONFIGURING OPENSIFT TO ENABLE SERVICE DISCOVERY

As a 3scale administrator, you have two options to configure Service Discovery: with or without an Open Authorization (OAuth) server.

If you configure 3scale Service Discovery with an OAuth server, this is what happens when a user signs in to 3scale:

- The user is redirected to the OAuth Server.

- If the user is not already logged in to the OAuth Server, the user is prompted to log in.
- If it is the first time that the user implements 3scale Service Discovery with SSO, the OAuth server prompts for authorization to perform the relevant actions.
- The user is redirected back to 3scale.

To configure Service Discovery with an OAuth server, you have the following options:

- [Configuring Service Discovery with an OpenShift OAuth server](#)
- [Configuring Service Discovery with an RH-SSO server \(Keycloak\)](#)

If you [configure Service Discovery without an OAuth server](#), when a user signs in to 3scale, the user is not redirected. Instead, the 3scale Single Service Account provides a seamless authentication to the cluster for the Service Discovery. All 3scale tenant administration users have the same access level to the cluster while discovering API services through 3scale.

## 29.4. CONFIGURING SERVICE DISCOVERY WITH AN OPENSIFT OAUTH SERVER

As a 3scale system administrator, allow users to individually authenticate and authorize 3scale to discover APIs by using OpenShift built-in OAuth server.

### Prerequisites

- You must deploy 3scale 2.11 to an OpenShift cluster (version 3.11 or later).
- To deploy 3scale to OpenShift, you need to use [3scale-amp-openshift-templates](#).
- 3scale users that want to use Service Discovery in 3scale must have access to the OpenShift cluster.

### Procedure

1. Create an OpenShift OAuth client for 3scale. For more details, see the [OpenShift Authentication documentation](#). In the following example, replace **<provide-a-client-secret>** with a secret that you generate and replace **<3scale-master-domain-route>** with the URL to access the 3scale Master Admin Portal.

```
$ oc project default
$ cat <<-EOF | oc create -f -
kind: OAuthClient
apiVersion: v1
metadata:
  name: 3scale
secret: "<provide-a-client-secret>"
redirectURIs:
  - "<3scale-master-domain-route>"
grantMethod: prompt
EOF
```

2. Open the 3scale Service Discovery settings file:

```
$ oc project <3scale-project>
$ oc edit configmap system
```

3. Configure the following settings:

```
service_discovery.yml:
  production:
    enabled: true
    authentication_method: oauth
    oauth_server_type: builtin
    client_id: '3scale'
    client_secret: '<choose-a-client-secret>'
```

4. Ensure that users have proper permissions to view cluster projects containing discoverable services.

To give an administrator user, represented by `<user>`, the view permission for the `<namespace>` project containing a service to be discovered, use this command:

```
oc adm policy add-role-to-user view <user> -n <namespace>
```

5. After modifying **configmap**, you must redeploy the **system-app** and **system-sidekiq** pods to apply the changes.

```
oc rollout latest dc/system-app
oc rollout latest dc/system-sidekiq
```

6. Check the status of the rollout to ensure it has finished:

```
oc rollout status dc/system-app
oc rollout status dc/system-sidekiq
```

### Additional resources

- For more information about OpenShift OAuth tokens, see [Configuring the internal OAuth server](#).

## 29.5. CONFIGURING SERVICE DISCOVERY WITH AN RH-SSO SERVER (KEYCLOAK)

As a system administrator, allow users to individually authenticate and authorize 3scale to discover services by using [Red Hat Single Sign-On for OpenShift](#).

For an example about configuring OpenShift to use the RH-SSO deployment as the authorization gateway for OpenShift, you can refer to this [workflow](#).

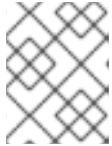
### Prerequisites

- You must deploy 3scale 2.11 to an OpenShift cluster (version 3.11 or later).
- To deploy 3scale to OpenShift, you need to use [3scale-amp-openshift-templates](#).

- 3scale users that want to use Service Discovery in 3scale must have access to the OpenShift cluster.

## Procedure

1. Create an OAuth client for 3scale in Red Hat OAuth server (Keycloak).



### NOTE

In the client configuration, verify that the **username** maps to **preferred\_username**, so that OpenShift can link accounts.

2. Edit 3scale Service Discovery settings.

```
$ oc project <3scale-project>
$ oc edit configmap system
```

3. Verify that the following settings are configured, where `<the-client-secret-from-Keycloak>` is the value that Keycloak generated automatically when you created the OAuth client.

```
service_discovery.yml:
  production:
    enabled: true
    authentication_method: oauth
    oauth_server_type: rh_sso
    client_id: '3scale'
    client_secret: '<the-client-secret-from-Keycloak>'
```

4. Make sure that users have proper permissions to view cluster projects containing discoverable services.  
For example, to give `<user>` view permission for the `<namespace>` project, use this command:

```
oc adm policy add-role-to-user view <user> -n <namespace>
```

5. After modifying **configmap**, you must redeploy the **system-app** and **system-sidekiq** pods to apply the changes.

## Additional resources

- Token lifespan: By default, session tokens expire after one minute, as indicated in [Keycloak - Session and Token Timeouts](#). However, it is recommended to set the timeout to an acceptable value of one day.

## 29.6. CONFIGURING SERVICE DISCOVERY WITHOUT AN OAUTH SERVER

To configure the 3scale Service Discovery without an OAuth server, you can use 3scale Single Service Account to authenticate to OpenShift API service.

### Prerequisites

- You must deploy 3scale 2.11 to an OpenShift cluster (version 3.11 or later).

- To deploy 3scale to OpenShift, you need to use [3scale-amp-openshift-templates](#).
- 3scale users that want to use Service Discovery in 3scale must have access to the OpenShift cluster.

### Procedure

1. Verify that the 3scale project is the current project.

```
$ oc project <3scale-project>
```

2. Open the 3scale Service Discovery settings in an editor.

```
$ oc edit configmap system
```

3. Verify that the following settings are configured.

```
service_discovery.yml:
  production:
    enabled: <%= cluster_token_file_exists = File.exists?(cluster_token_file_path =
'/var/run/secrets/kubernetes.io/serviceaccount/token') %>
    bearer_token: "<%= File.read(cluster_token_file_path) if cluster_token_file_exists %>"
    authentication_method: service_account
```

4. Provide the 3scale deployment **amp** service account with the relevant permissions to view projects containing discoverable services by following one of these options:

- Grant the 3scale deployment **amp** service account with **view** cluster level permission.

```
oc adm policy add-cluster-role-to-user view system:serviceaccount:<3scale-project>:amp
```

- Apply a more restrictive policy as described in [OpenShift - Service Accounts](#).

## 29.7. IMPORTING DISCOVERED SERVICES

From the OpenShift cluster, import a new API service that conforms to the OpenAPI Specification. This API is managed with 3scale.

### Prerequisites

- The OpenShift administrator has configured Service Discovery for the OpenShift cluster. For example, the OpenShift administrator must have enabled 3scale discovery by editing the Fuse Online custom resource to specify the URL for their 3scale user interface.
- The 3scale administrator has configured the 3scale deployment for Service Discovery as described in [About Service Discovery](#).
- The 3scale administrator has granted your 3scale user or service account (depending on the configured authentication mode) the necessary privileges to view the API service and its namespace. For more details, you can see [Authorizing 3scale access to an OpenShift project](#).
- The API has the correct annotations that enable Service Discovery, as described in [Criteria for a discoverable service](#).

- The API service is deployed on the same OpenShift cluster where 3scale is installed.
- You know the API's service name and its namespace (OpenShift project).

### Procedure

1. Log in to the 3scale Admin Portal.
2. From **APIs** on the Admin Portal Dashboard, click **Create Product**
3. Click **Import from OpenShift**.
  - If the OAuth token is not valid, the OpenShift project administrator should authorize access to the 3scale user as described in [Authorizing 3scale access to an OpenShift project](#).
4. In the **Namespace** field, specify or select the OpenShift project that contains the API, for example **fuse**.
5. In the **Name** field, type or select the name of an OpenShift service within that namespace, for example **i-task-api**.
6. Click **Create Product**.
7. Wait for the new API service to be asynchronously imported into 3scale. A message appears in the upper right section of the Admin Portal: **The service will be imported shortly. You will receive a notification when it is done.**

### Additional resources

- See the [Red Hat 3scale API Management documentation](#) for information about managing the API

## 29.8. AUTHORIZING 3SCALE ACCESS TO AN OPENSIFT PROJECT

As an OpenShift project administrator, you can authorize a 3scale user to access a namespace when the OAuth token is not valid.

### Prerequisites

- You need to have the credentials as an OpenShift project administrator.
- The OpenShift administrator has configured Service Discovery for the OpenShift cluster. For example, for Fuse Online APIs, the OpenShift administrator must set the Fuse Online service's **CONTROLLERS\_EXPOSE\_VIA3SCALE** environment variable to **true**.
- The 3scale administrator has configured the 3scale deployment for Service Discovery as described in [Criteria for a discoverable service](#).
- You know the API service name and its namespace of the OpenShift project.
- The API service is deployed on the same OpenShift cluster where 3scale is installed.
- The API has the correct annotations that enable Service Discovery, as described in [Criteria for a discoverable service](#).

### Procedure



1. Click the *Authenticate to enable this option* link.
2. Log in to OpenShift using the namespace administrator credentials.
3. Authorize access to the 3scale user, by clicking **Allow selected permissions**.

### Additional resources

- See the [Red Hat 3scale API Management documentation](#) for information about managing the API.

## 29.9. UPDATING SERVICES

You can update (refresh) an existing API service in 3scale with the current definitions for the service in the cluster.

### Prerequisites

- The service was previously imported from the cluster, as described in [Importing discovered services](#).

### Procedure

1. Log in to 3scale Admin Portal.
2. Navigate to the Overview page of the API product.
3. Click the **Refresh** link, next to Source: OpenSource.
4. Wait for the new API service to be asynchronously imported into 3scale.

## PART VI. MULTITENANCY

## CHAPTER 30. MULTITENANCY

Red Hat 3scale API Management allows multiple independent instances of 3scale accounts to exist on a single on-premises deployment. Accounts operate independently from one another, and cannot share information among themselves.

### 30.1. MASTER ADMIN PORTAL

A master administrator monitors and manages the 3scale accounts through the Master Admin Portal and API endpoints. Similar to the standard [Admin Portal](#), the Master Admin Portal contains information about all the accounts of a deployment, and allows for administration of accounts and users through a unique account page.

For details on account administrator operations, refer to the [Accounts](#) guide.

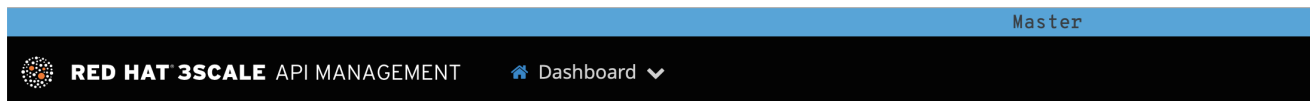
#### 30.1.1. Accessing the Master Admin Portal

To access the Master Admin Portal, you need to use the credentials and URL specifically defined for the Master Admin Portal during the [on-premises installation](#) process.

The Master Admin Portal URL consists of the **MASTER\_NAME** (*master* by default in the template) and the **WILDCARD\_DOMAIN**:

```
<MASTER_NAME>.<WILDCARD_DOMAIN>
```

You can identify the Master Admin Portal by the **Master** flag.



#### 30.1.2. Adding an account through the Master Admin Portal

To add an account through the Master Admin Portal, follow these steps:

1. Log in to the [Master Admin Portal](#).
2. Navigate to **Accounts**.
3. Click **Create**.
4. Indicate the required information for the user:
  - a. Username
  - b. Email
  - c. Password
  - d. Password confirmation
5. Indicate the required information for the organization:
  - a. Organization/Group Name

## 6. Click **Create**.

After these steps, Red Hat 3scale creates an account subdomain for your account based on the **Organization/Group Name** field. Additionally, you can see a page containing the details of the account you created.

### 30.1.3. Creating a single gateway with the Master Admin Portal

With the Master Admin Portal, you can create a single gateway for all tenants by configuring the `THREESCALE_PORTAL_ENDPOINT` environment variable. This is similar to the Hosted APIcast in the Hosted 3scale (SaaS), where the default **apicast-staging** and **apicast-production** gateways deployed with the OpenShift template are configured in this way.

To create a single gateway with the Master Admin Portal, follow these steps:

1. Fetch the value of `ACCESS_TOKEN` out of the **system-master-apicast** secret in your 3scale project. Alternatively, you can [create new access tokens](#) in the Master Admin Portal.
2. Use the following command when you deploy APIcast:

```
THREESCALE_PORTAL_ENDPOINT="https://<ACCESS_TOKEN>@<public url to master admin portal>/master/api/proxy/configs"
```

- The end of the url looks like **/master/api/proxy/configs**, this is because *master* holds the **configs** in a different endpoint compared to the default **/admin/api/services.json**.

## 30.2. MANAGING ACCOUNTS

You can manage accounts through the Master Admin Portal or through API calls.

### 30.2.1. Managing accounts through the Master Admin Portal

To manage the accounts through the Master Admin Portal, you need to do the following:

1. Log in to the [Master Admin Portal](#).
2. Navigate to the **Accounts** page.
3. Select the group or organization you want to manage.

From the **Accounts** page, you can perform administrative actions, such as impersonating an admin account or suspending an account. You can also manage the following account attributes:

- Applications
- Users
- Invitations
- Group Memberships
- Organization/Group Name

### 30.2.2. Managing accounts through API calls

You can manage accounts through the Master Admin API calls. For information on these calls, refer to the **Master API** section, by clicking the question mark (?) icon located in the upper-right corner of the Master Admin Portal, and then choosing **3scale API Docs**.

#### Master API

Trigger Billing by Account	/master/api/providers/{provider_id}/accounts/{account_id}/billing_jobs.xml	POST
Trigger Billing	/master/api/providers/{provider_id}/billing_jobs.xml	POST
Tenant Create	/master/api/providers.xml	POST
Tenant Update	/master/api/providers/{id}.xml	PUT
Tenant Delete	/master/api/providers/{id}.xml	DELETE

### 30.3. UNDERSTANDING MULTITENANCY SUBDOMAINS

As a result of multiple accounts existing under the same OpenShift cluster domain, individual account names prepend the OpenShift cluster domain name as subdomains. For example, the route for an account named **user** on a cluster with a domain of **example.com** appears as:

```
user.example.com
```

A standard multitenant deployment will include:

- A master admin user
- A master admin portal route, defined by the **MASTER\_NAME** parameter:

```
<MASTER_NAME>.<WILDCARD_DOMAIN>
```

- An account admin user
- An account admin portal route, defined by the **TENANT\_NAME** parameter:

```
<TENANT_NAME>-admin.<WILDCARD_DOMAIN>
```

- A developer portal route for the account:

```
<TENANT_NAME>.<WILDCARD_DOMAIN>
```

- Routes for the production and staging embedded APIcast gateway:

```
<API_NAME>-<TENANT_NAME>-apicast-staging.<WILDCARD_DOMAIN>
<API_NAME>-<TENANT_NAME>-apicast-production.<WILDCARD_DOMAIN>
```

This example illustrates the output users and routes of a standard multitenant deployment of 3scale:

```
----
--> Deploying template "3scale-project/3scale-api-management" for "amp.yml" to project
project
```

## 3scale API Management

### 3scale API Management main system

```

Login on https://user-admin.3scale-project.example.com as admin/xXxYyz123
...
* With parameters:
* ADMIN_PASSWORD=xXxYyz123 # generated
* ADMIN_USERNAME=admin
* TENANT_NAME=user
...
* MASTER_NAME=master
* MASTER_USER=master
* MASTER_PASSWORD=xXxYyz123 # generated
...
--> Success
Access your application via route 'user-admin.3scale-project.example.com'
Access your application via route 'master-admin.3scale-project.example.com'
Access your application via route 'backend-user.3scale-project.example.com'
Access your application via route 'user.3scale-project.example.com'
Access your application via route 'api-user-apicast-staging.3scale-project.example.com'
Access your application via route 'api-user-apicast-production.3scale-project.example.com'
Access your application via route 'apicast-wildcard.3scale-project.example.com'
...
----
```

Additional accounts [added by the master admin](#) will be assigned a subdomain based on their names.

## 30.4. DELETING TENANT ACCOUNTS

### 30.4.1. Deleting an account via the Admin Portal

With this procedure, accounts are scheduled for deletion and will be deleted after 15 days. During the time it is scheduled for deletion:

- Users cannot log in to the account.
- The account can not be edited; but the master can resume the account to the *approved* status.

Additionally, the domains of the tenant (admin domain and developer portal) are not available, similar to a real deletion.

#### Prerequisites:

- Log in to your [master admin account](#).

#### Procedure

1. To see the list of accounts, navigate to **Accounts**.
2. Click the account you want to delete.
3. Click **Edit**, next to the account's name.

4. In the accounts details page, click the **Delete** icon.
5. Confirm the deletion.

### 30.4.2. Deleting a tenant via the console

If you want to delete the account with an immediate effect, you can do so via the console:

1. Open the console with these commands:

```
oc rsh -c system-master "$(oc get pods --selector deploymentconfig=system-app -o name)"
bundle exec rails console
```

2. Delete immediately with these lines:

```
tenant = Account.find(PROVIDER_ID)
tenant.schedule_for_deletion!
DeleteAccountHierarchyWorker.perform_later(tenant)
```

This is how each line works:

- Line 1: finds the account and saves it in the variable **tenant**.
- Line 2: schedules the account for deletion. This is only necessary if you have not scheduled the deletion through the Admin Portal.
- Line 3: deletes the tenant in a background process only if you have scheduled the account for deletion or it is suspended. Deletion will not proceed if the account is in *approved* status.

## 30.5. RESUMING TENANT ACCOUNTS

Resuming a tenant account implies restoring an account scheduled for deletion. You can resume a tenant account up to 15 days after you have scheduled it for deletion.

After resuming an account:

- All previous apps exist.
- All historical stats remain.
- All tokens that should be valid are valid again.
- Apps start authorizing again.

#### Prerequisites:

- Log in to your master admin account.

#### Procedure

1. To see the list of accounts, navigate to **Accounts**.
2. Click the account you want to delete.
3. Under the account details, click **Resume**.

4. Click **Ok** to confirm you want to resume the account.



## PART VII. ANALYTICS

## CHAPTER 31. IMPLEMENTING 3SCALE API ANALYTICS TO MANAGE AND OPTIMIZE API ACCESS

Implementing 3scale API analytics to manage and optimize API access enables you to track items such as usage trends over time. Knowing how your API is used is a crucial step for managing traffic, provisioning for peaks, and identifying users who send the most requests to the API.

3scale gathers API analytics for methods and metrics that you can define at these levels:

- **Product:** *Hits* is the built-in metric that tracks traffic to your API. You can create additional metrics and designate methods in your API for which to capture analytics.
- **Backend:** 3scale registers methods and metrics in the backend as if they belong to each product using the API backend. You can set limits and pricing rules for the backend-level metrics in the Application Plans defined at the Product level.
- **Application:** You can obtain analytics reports for each application created in 3scale.

### Prerequisites

- You completed the [Getting Started instructions](#).  
By using the *Getting Started* guide, you use one of the existing 3scale code plugins to perform an integration.
- Alternatively, follow a similar flow with other integration methods. See the [Operating APIcast](#) chapter of the documentation to learn more about the available integration options.

### 31.1. 3SCALE API METRICS AND METHODS THAT CAPTURE API USE

3scale acts as an infinitely scalable data repository for your API product statistics. You can use metrics and methods to capture API product statistics so that you have the information necessary to optimally manage access to you API. For example:

- *Hits/transactions:* calls to the API product. Hits are included by default as metrics on all APIs. Hits can be overall calls to the API product or broken out into individual methods on the API product.
- *Data transfer:* quantity of MB/GB of data uploaded and downloaded via the API product
- *CPU hours:* compute time (or some other internal resource) associated with calls to the API product
- *Results returned:* count of the number of records or data objects being returned
- *Disk storage:* total disk storage in use by an account

You can track more metrics that are relevant to your API product. 3scale can track an arbitrary number of metrics and methods, as long as it is a countable quantity that can be incremented over time.

Once you have chosen the metrics you want to use, register them in the Admin Portal by using the procedure described in [Adding metrics to products and backends](#).

You can add metrics and methods to the product or backend you have selected. Provide them with a friendly name and a system name, which 3scale uses in the plugin configuration. For more details about creating methods and metrics, see [Designating methods and adding metrics for capturing usage details](#).

## 31.2. CONFIGURING 3SCALE PLUGINS TO CAPTURE API METRICS

When 3scale is ready with the names of the metrics you want to track, you need to configure your plugin to report those metrics. The precise manner of doing this depends on the plugin or integration method in use. By default, the plugins report the *Hits*, API transactions, metric only.

### Prerequisites

- You must have already configured metrics for products and/or backends.

### Procedure

1. Pass the appropriate metric/method names to the plugin as determined by the incoming API call.  
The metric/method value and the increment required is an argument of authorize and/or report methods the plugin exposes.
2. Use the *System name* method in the metric to report traffic for a specific API method.  
This automatically increments the counter both for the method reported and the *Hits* metric.
3. In the Admin Portal, under **Account Settings > Integrate → 3scale API Docs** report the traffic using the *3scale Service Management API*.

For additional information about different endpoints, refer to the *3scale API Documentation*, available in the Admin Portal under **Account Settings > Integrate → 3scale API Docs**

## 31.3. VIEWING ANALYTICS FOR 3SCALE API BACKENDS

You can view the traffic data for a given API configured as a backend. The *Traffic* page shows the hits to the endpoint.

### Prerequisites

- You must have already created an API backend. See [Creating backends for your products](#)
- You must have added the backend to an API product. See [Adding backends to your products](#)

### Procedure

1. From the Admin Portal dashboard, navigate to **[Your\_backend\_name] > Analytics**.
2. Choose the time frame for the report. You can choose to show the last 24 hours, 7 days, 30 days, 12 months; or indicate a period of time.
3. Optional: Select an analytic other than *Hits*, either a method or another top level metric.
  - You will see the results in the chart. If there is no traffic to report, you will see the following message: *There is no data available for the selected period*
4. Click the **Download CSV** link to download the data in a CSV file.

## 31.4. SENDING TEST REQUESTS TO CONFIRM PLUGIN CONFIGURATION FOR CAPTURING API METRICS

Establish a connection between the API and 3scale so that you can send traffic to the API product and watch it register on the API Analytics dashboard.

## Prerequisites

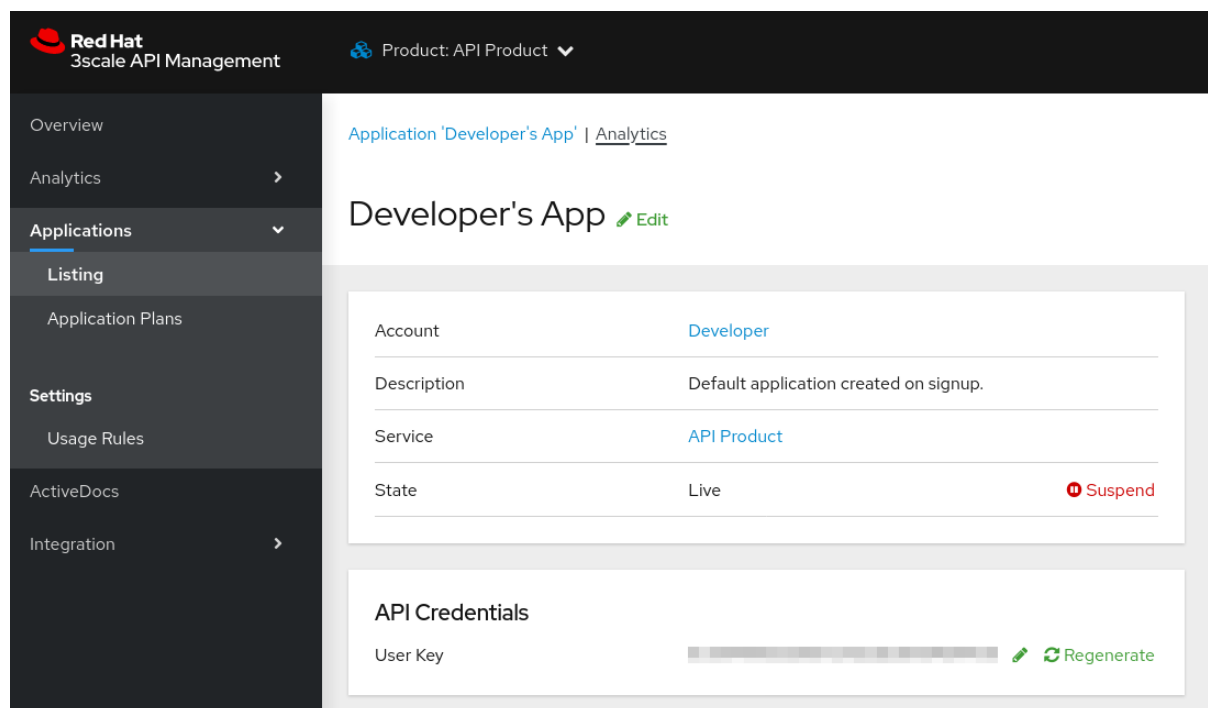
- A 3scale developer account
- A 3scale application with API credentials

## Procedure

1. Navigate to **Audience > Applications > Listing** to see the list of existing applications.
2. Select an application by clicking its name.
3. Find the *API Credentials* of the selected application. The credentials depend on the selected authentication type and can be a user key (API key), an application ID and application keys, or a client ID and client secret.

For more information about the available authentication modes, see the [Authentication patterns](#).

Figure 31.1. API credentials



4. Use these keys to make calls to your API in the usual way. For example, from the command line using **cURL** or from the browser for API endpoints using the **GET** method. The precise calls to make depend on the structure of the methods on your API product. Traffic from these calls appears in the Analytics section for your API product.

## Next steps

By default, the usage statistics are visible to the API provider through the Admin Portal and to the developers who created applications through the Developer Portal. Each developer can see only the usage statistics for their own applications. You can hide the analytics views from the Developer Portal to further control who sees them. See [Creating the Developer Portal](#) for more information about customizing the Developer Portal.

Besides the usage graphs in the Analytics section, you can access analytics data with the Analytics API. The 3scale Analytics API is a flexible way to extract all analytics data for your API product in either XML or JSON format.

## 31.5. TROUBLESHOOTING TECHNIQUES WHEN ANALYTICS ARE MISSING FOR 3SCALE APIS

If traffic does not display on the usage charts in the `[Your_product_name] > Analytics > Traffic`, perform the following checks.

- **Are authorize/report calls responding correctly?**  
All plugins call the 3scale Service Management API, which has predetermined response codes. Authorize calls for valid keys should return responses with HTTP code **200**. Report calls should respond with code **202**.
- **Are there errors in the integration error console?**  
The log of integration errors detected by 3scale can be found in `[your_product_name] > Analytics > Integration Errors`.
- **Are the correct metric and method names being used?**  
The most common reason for failure is that the method and metric names passed in report calls do not correspond to those created in the API settings of your Admin Portal. Check that you are using the correct **system names** for each metric/method.
- **Are mapping rules properly mapped to the metrics?**  
If mapping rules are not properly mapped to the metrics, you may not see data in the Analytics. Check that the mapping rules are correctly mapped to the metrics, both for API products and API backends:
  - Products: From the Admin Portal Dashboard, navigate to `[Your_product_name] > Integration > Methods & Metrics`.
  - Backends: From the Admin Portal Dashboard, navigate to `[Your_backend_name] > Methods & Metrics`.

## CHAPTER 32. EXPORTING 3SCALE API ANALYTICS BEYOND BUILT-IN CAPABILITIES

Create scripts that extend the capabilities of built-in 3scale analytics so that you can retrieve information that is not provided by default in the features.

By using Account Management and Analytics API (Enterprise only), you can create scripts to retrieve information that you need in your preferred format. The use case described here can be used to help you in your own scenario to get the data needed out of 3scale.

### Reasons for customized scripts

3scale continuously improves the features available on your API Dashboard. However, you may be ahead of our development plans and have a very specific need not yet supported.

To satisfy the needs for API management, the 3scale Admin Portal provides the tools you need to access all your data. It takes resources to write the scripts, however customized scripts gives you total flexibility and control to implement various use cases.

### 32.1. EXAMPLE OF USING 3SCALE TO EXTRACT DATA ABOUT APPLICATION USE

A customer was in the process of onboarding thousands of new developers per week. This customer could tackle some aspects of the onboarding because 3scale provides automated necessities such as key provisioning, sign-up workflows, and email communication. There was, however, something that was not possible to do with 3scale, which was quite important for them.

Since the customer was onboarding many people, the company needed a straight-forward way to classify the new developers based on their engagement with APIs so that their operations and marketing teams could interact with the new developers more effectively. At least at the required level of detail, such a feature was not yet available in the built-in analytics tools provided by 3scale. However, it was possible to extract data using the 3scale Account and Analytics API because all the data was available in the system.

#### Example: Customer requirements

They would like to know how many new developers have signed up for the free evaluation plan in the last 10 days, split up different ways.

First, they wanted to know how many developers signed up but never used their API.

Second, they wanted to split the developers that had used their API into two groups:

- Developers that used it for a period of time – say the first half of the 10 days – and then stopped using the API. These developers tried it out, but became inactive.
- Developers that have been using the API consistently. For those, they would like to know the percent growth (or decline).

This information is available in 3scale built-in analytics. The problem is that there is no view to show it aggregated, which makes the whole experience quite cumbersome.

### 32.2. EXTRACTING 3SCALE APPLICATION ANALYTICS IN CUSTOM PROCEDURES

To extract application analytics, start by working with ActiveDocs. 3scale ActiveDocs is available in your Admin Portal, under **Account Settings > Integrate > 3scale API Docs**. 3scale has all of its APIs available as ActiveDocs so that you can try them from your browser. This allows you to find the request that best serves your needs, get the request (curl-like) and get a grasp of the response. The following figure provides an ActiveDocs example:

The screenshot shows the ActiveDocs interface for the 'Application List' API. The title is 'Application List (all services)' and the URL is '/admin/api/applications.xml'. The description states: 'Returns the list of applications across all services. Note that applications are scoped by service. Results can be paginated.' Below the description is a table of parameters:

PARAMETER	VALUE	DESCRIPTION
provider_key	4e8d8fc2ea7cf03ff81423784952b1198	Your api key with 3scale (also known as provider key).
page	1	Page in the paginated list. Defaults to 1.
per_page	500	Number of results per page. Default and max is 500.
active_since		filter date
inactive_since		filter date
service_id		filter by service

Below the table are buttons for 'Send Request' and 'HIDE RESPONSE'. The 'Request' section shows the following curl command:

```
curl -v -X GET "https://enterprisedemo-admin.3scale.net/admin/api/applications.xml?provider_key=4e8d8fc2ea7cf03ff81423784952b1198&page=1&per_page=500"
```

The 'Response Body' section shows the following XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<applications per_page="500" total_entries="1700" total_pages="4" current_page="1">
  <application>
    <id>1409607484442</id>
    <created_at>2012-10-25T08:55:42Z</created_at>
    <updated_at>2015-12-07T13:54:39Z</updated_at>
    <state>live</state>
    <user_account_id>2445579995682</user_account_id>
    <end_user_required>false</end_user_required>
    <service_id>1006371742031</service_id>
    <user_key>930227b30f52ca2e06b13c18f3153c1</user_key>
    <provider_verification_key>4026533af7915a76a84c17bd4f350a64</provider_verification_key>
    <plan custom="false" default="false">
      <id>2357355464251</id>
    </plan>
  </application>
</applications>
```

This is the ActiveDocs for the API request that fetches all applications for which the script will extract analytics.

- After you have done the research with the ActiveDocs, specify the request in your scripting language of choice. The example uses Ruby.
- Repeat until you have a script that does exactly what you need. For the example of the extended analytics, the script is [available as a gist](#). You can try it out in your account.

ActiveDocs lets you quickly understand what the API is able to do. Then, it is a matter of finding which 3 or 4 requests are needed for the task you want to do and putting a script together.

The following procedure shows the steps that achieved the custom analytics that the example customer wanted.

## Procedure

1. Retrieve the full list of applications. This operation requires pagination.

```
def api_call_applications_list(domain, provider_key)
  done = false
  res = Array.new
  page = 1

  while !done
    url = "https://#{domain}/admin/api/applications.xml?provider_key=#{provider_key}&page=#{page}&per_page=100"
    page += 1
  end
end
```

```

response = RestClient.get url
raise Exception.new("Wrong response code (#{response.code}) in request #{url}") if
response.code!=200
document = Nokogiri::XML(response.to_str) done =
document.xpath("applications/@current_page").text ==
document.xpath("applications/@total_pages").text
document.xpath("//application").each do |item|
  app = Hash.new
  app["created_at"] = DateTime.parse(item.xpath("created_at").text)
  app["plan_name"] = item.xpath("plan/name").text
  app["service_id"] = item.xpath("plan/service_id").text
  app["account_id"] = item.xpath("user_account_id").text
  app["id"] = item.xpath("id").text
  res << app
end
end
return res
end

```

2. Filter the applications that do not meet the criteria, i.e. the plan must be "evaluation" and they have to be newer than 10 days.

```

def filter_applications(domain, provider_key, plan_name, num_of_days)
res = api_call_applications_list(domain, provider_key)
res.each do |item|
  res.delete(item) if item["plan_name"] != plan_name
  res.delete(item) if item["created_at"] > (DateTime.now - num_of_days)
end
return res
end

```

3. For each application that meets the criteria, get its usage, which is the number of hits the application has had in the last 10 days.

```

def api_call_application_usage(domain, provider_key, application_id, metric, from, to,
granularity)
url = "https://#{domain}/stats/applications/#{application_id}/usage.xml?provider_key=#
{provider_key}&metric_name=#{metric}&since=#{from}&until=#{to}&granularity=#
{granularity}"
response = RestClient.get url
raise Exception.new("Wrong response code (#{response.code}) in request #{url}") if
response.code!=200
document = Nokogiri::XML(response.to_str)
return document.xpath("//usage/data/values").text.split(",")
end

```

4. Cross-reference the applications to the accounts, because the information for the developers is stored in the account object.

```

def api_call_account_read(domain, provider_key, account_id)
url = "https://#{domain}/admin/api/accounts/#{account_id}.xml?provider_key=#
{provider_key}"
response = RestClient.get url
raise Exception.new("Wrong response code (#{response.code}) in request #{url}") if
response.code!=200

```



```
document = Nokogiri::XML(response.to_str)
account = Hash.new
account["email"] = document.xpath("//users/user/email").text
account["name"] = document.xpath("//users/user/first_name").text + " " +
document.xpath("//users/user/last_name").text
return account
end
```

5. Put everything together to complete the script. You have a script that gets the information that was not yet available in 3scale's built-in analytics. You can also get the [full script as a gist](#).

## CHAPTER 33. VIEWING 3SCALE BUILT-IN TRAFFIC ANALYTICS FOR APPLICATIONS

To identify accounts that need support and operations that warrant enhancements, view visual information on the traffic of an application. Each application using an API has a traffic trace in the 3scale system, which can be viewed from the Admin Portal as well as recovered by an API.

### Prerequisites

- A 3scale developer account
- A 3scale application with API credentials

### Procedure

1. Navigate to the application for which you want to view analytics.  
You can find the application from **Audience > Accounts > Listing** or **Audience > Applications > Listing** under the *Applications* tab. Alternatively, you can search for an application by following the steps described in the [finding applications tutorial](#).
2. After you have located the application, you will see an overview page with information about the application as shown in the following image:

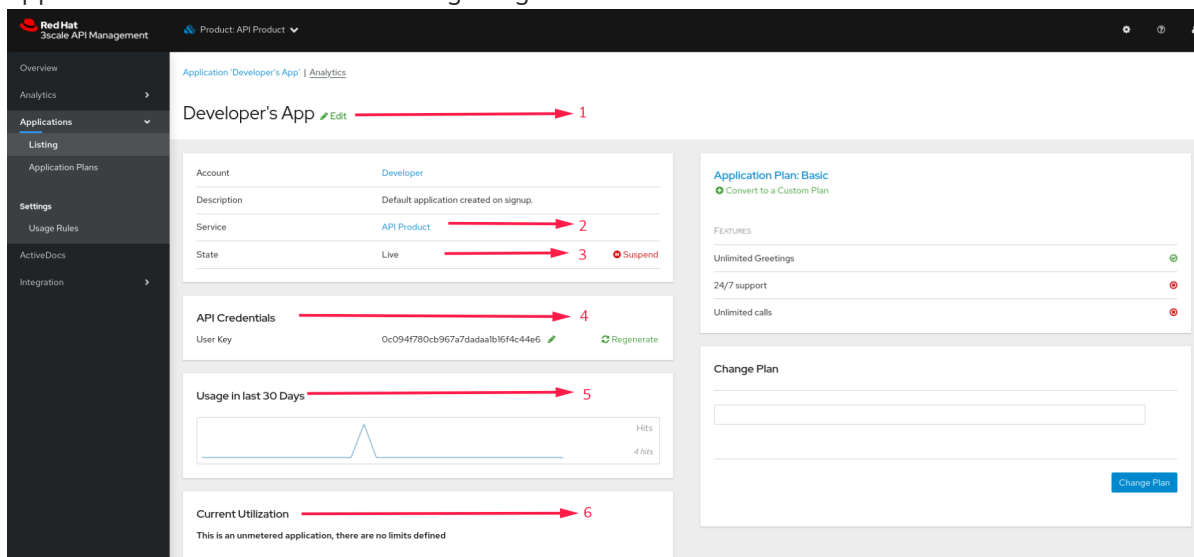


Table 33.1. The items labeled in the image correspond to the following information

Number	Application information
1	The name of the application given by the developer
2	Meta data captured for the application (learn how to set which data to capture in the advanced section)
3	The status of the application – is it live or suspended?
4	The current API identifiers, keys, and certificates that the application has. The view varies depending on what authentication method you are using to integrate the API into 3scale.

Number	Application information
5	A summary of traffic statistics for the application
6	Information on which application plan the application is on and which rate limits apply.

3. Click the **Analytics** link above the application name. The usage charts are displayed for the application. Controlling the metrics, methods, and time range allows you to check different types of data about the application.

## CHAPTER 34. SETTING UP AND EVALUATING THE 3SCALE RESPONSE CODES LOG FOR YOUR API

To see how your clients are using the API, and see in real time if your servers are running as expected, set up and use the response codes log in 3scale.

### Procedure

- Set the **APICAST\_RESPONSE\_CODES** environment variable to **1** or **true** when starting the APIcast gateway that uses [Docker](#) or an [OpenShift](#) deployment. This enables response code tracking.

### Example

When enabled, APIcast gateway captures the HTTP status code of API responses returned by the upstream service for authorized calls, and sends them to the Service Management API (in **authrep** call). Example:

```
https://su1.3scale.net/transactions/authrep.xml?service_token={SERVICE_TOKEN}&service_id={SERVICE_ID}&user_key={USER_KEY}&usage%5Bhits%5D=1&log%5Bcode%5D=200"
```

In this example, **log[code]=200** is sent, which means that the API responded with a **200** status code.

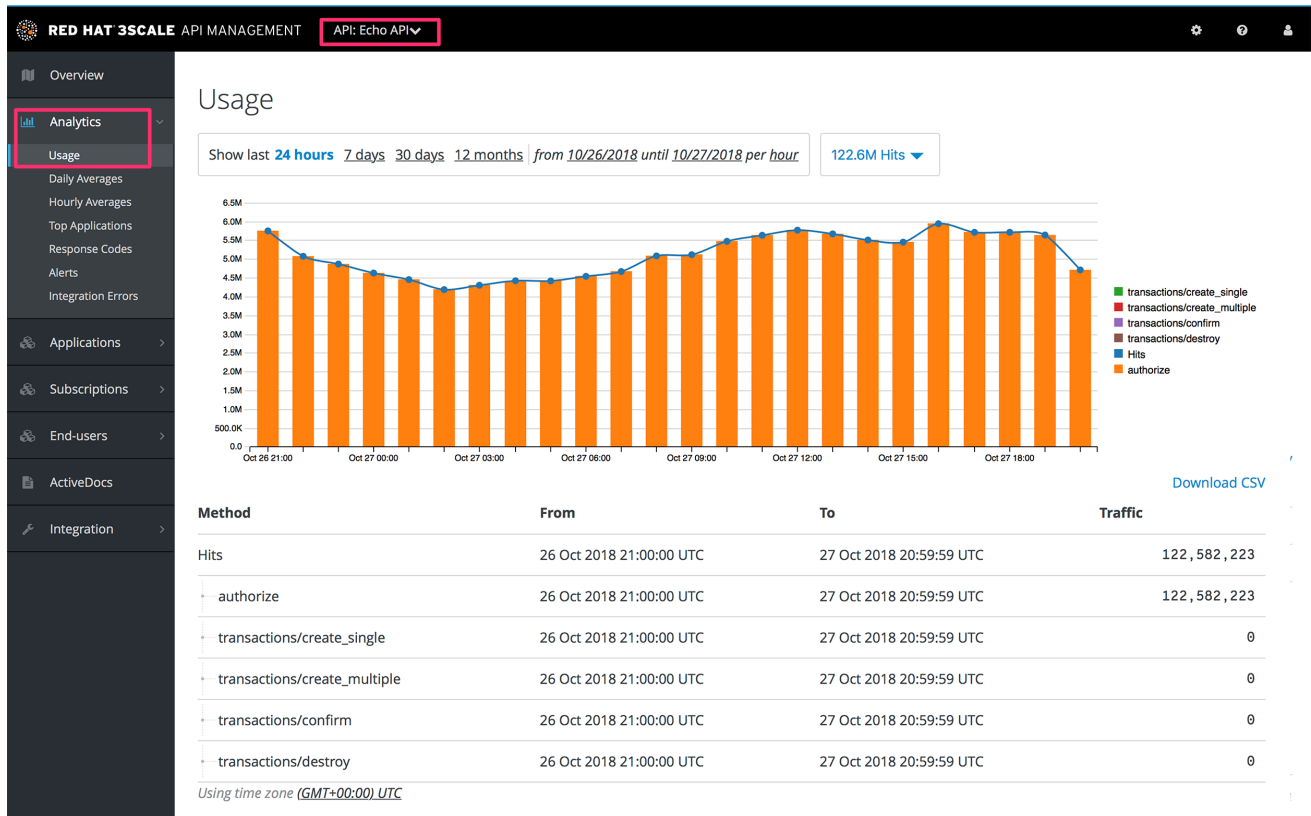
### Verification

To verify the integration, perform a call to your API product with valid 3scale credentials, and then verify that the call was correctly reported on the **Analytics > Usage** page.

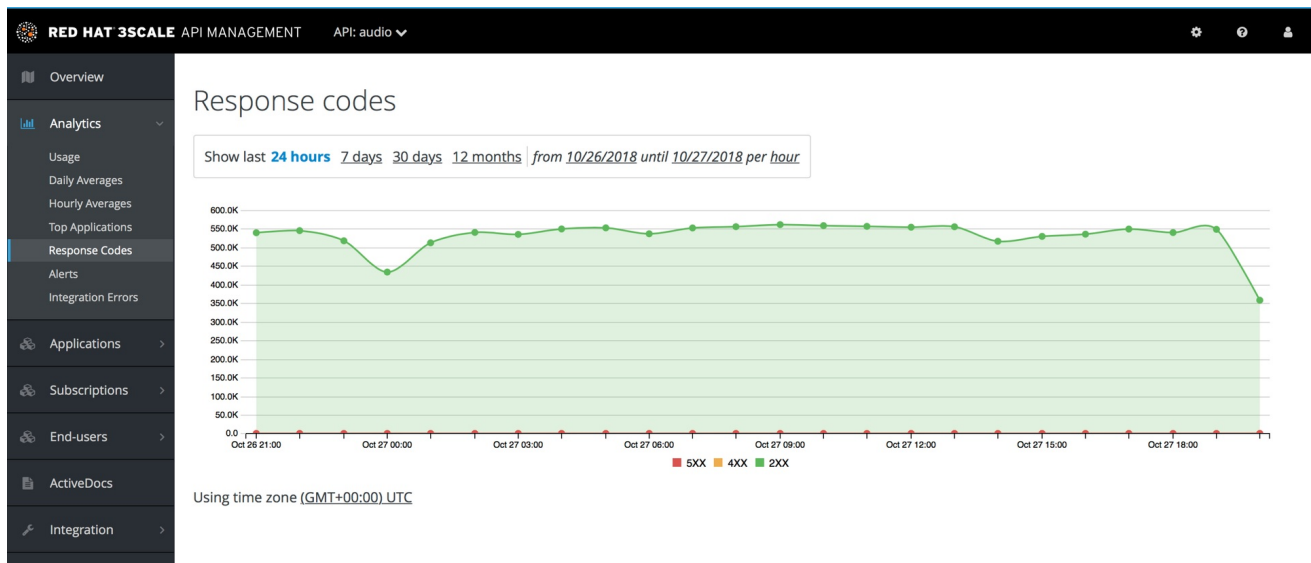


### NOTE

- Response Code Tracking is not intended to be an accurate count of all responses.
- The value of this view is to provide a visual representation of the trend over a period of time.
- Response Code Tracking and 3scale Auth Caching mode: **None** is not a supported combination.



If everything is going well so far, go to the **Analytics >Response codes** page. You should be able to see a graph with your latest traffic divided by colors, depending if the response was 2xx, 4xx, or 5xx.



The graph tool gives you the ability to view the history of response codes. You can also check the response code statistics for different periods of time and different levels of granularity. Click the time selection bar and define the time period and granularity that meets your needs.