Red Hat 3Scale 2-saas

管理ポータルガイド

Red Hat 3scale に関する諸要素の管理

Last Updated: 2023-02-03

Red Hat 3Scale 2-saas 管理ポータルガイド

Red Hat 3scale に関する諸要素の管理

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

Copyright © 2023 | You need to change the HOLDER entity in the en-US/Admin_Portal_Guide.ent file |.

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat 3scale の管理について説明します。

目次

前書き 6

多様性を受け入れるオープンソースの強化 7

パート I. アカウント設定 8

第1章 アカウント設定 9
1.1. 会社情報の追加 9
1.2. 希望するタイムゾーンの選択 9

第2章 3SCALE 管理ポータル用 RED HAT SINGLE SIGN ON 10
2.1. RH-SSO または AUTH0 メンバー認証の有効化 10
2.1.1. RH-SSO 使用の前提条件 10
2.1.2. Auth0 使用の前提条件 10
2.1.3. RH-SSO の有効化 10
2.2. 3SCALE での RH-SSO の使用 11
2.3. 3SCALE ログインの RH-SSO オプションへのリダイレクト 11
2.3.1. 前提条件 11
2.3.2. 必要な手順 11

第3章 ユーザーの招待および権限の管理 13
3.1. ユーザー管理画面への移動 13
3.2. 招待状の送信 13
3.3. 招待の応諾 13
3.4. 新しいユーザー権限の付与 14

第4章 通知 15
4.1. 通知のタイプ 15
4.2. 制約 15
4.3. メール通知へのサブスクライブ 15
4.4. WEB 通知 15

第5章 個人設定 17

第6章 トークン 18
6.1. アクセストークン 18
6.2. アクセストークンの作成 18
6.3. アクセストークンの使用 18
6.3.1. サービストークン 19

パート II. 開発者アカウントの管理 20

第7章 開発者の追加 21
7.1. 新たな開発者アカウントの作成 21
7.2. アプリケーションの設定 21
7.3. 開発者への通知 21

第8章 開発者の承認 22
8.1. メール通知からの承認 22
8.2. アカウントの承認 22
8.3. サービスの承認 22
8.4. アプリケーションの承認 23

第9章 アプリケーションプランの変更 25
9.1. アカウントプランの変更 25

9.2. サービスプランの変更 25
9.3. アプリケーションプランの変更 26
9.3.1. 補足情報 27

第10章 開発者への連絡 28
10.1. システム内の該当アプリケーションおよびアカウントの把握 28
10.2. 開発者への内部メッセージの送信 28
10.3. 他の手段による連絡 29

第11章 プランのカスタマイズ 30
11.1. アカウントの選択 30
11.2. アプリケーションの選択 30
11.3. アプリケーションプランのカスタマイズ 30
11.3.1. 補足情報 31

第12章 セルフサービスによるサインアップの有効化 32

第13章 アプリケーションの特定 33
13.1. 必要な情報の入手 33
13.2. アプリケーションの検索 33
13.3. アプリケーション情報へのアクセス 33

第14章 開発者の招待 35

第15章 開発者のサービスからのサブスクライブ解除 36
15.1. 1 人の開発者のサービスからのサブスクライブ解除 36
15.2. 複数開発者のサービスからのサブスクライブ解除 36

第16章 アプリケーションの一時中断 37
16.1. アプリケーションの特定 37
16.2. アプリケーションの無効化 37
16.3. 開発者への連絡 37

第17章 アプリケーションの削除 39

第18章 API の削除 40

パート III. アクセス制御 41

第19章 使用状況の詳細を把握するためのメソッドの指定およびメトリクスの追加 42
19.1. プロダクトおよびバックエンドへのメソッドの追加 42
19.2. プロダクトおよびバックエンドへのメトリクスの追加 43
19.3. メソッドおよびメトリクスをインポートするその他の方法 44
19.4. メソッドおよびメトリクスへのマッピングルールの追加 44
19.5. 関連情報 45

第20章 アプリケーションプラン 46
20.1. アプリケーションプランの作成方法 46
20.2. デフォルトのアプリケーションプランの設定 46

第21章 有料プランのプロビジョニング 47
21.1. 課金モデルの決定 47
21.2. アプリケーションプランへの課金ルールの設定 47
21.3. 追加課金レイヤーの作成 48
21.4. 有料プランのプロビジョニング 48
21.5. その他の参考情報 49

第22章 流量制御のプロビジョニング **50**
22.1. アプリケーションプランの設定 50
22.2. 流量制御の設定 50
22.3. 新しい流量制御の適用 50
22.4. 補足情報 50

パート IV. 請求 **51**

第23章 請求設定の定義 **52**
23.1. MODE (CHARGING & GATEWAY) 52
23.2. CHARGING ENABLED (CHARGING & GATEWAY) 52
23.3. CURRENCY (CHARGING & GATEWAY) 52
23.4. INVOICE FOOTNOTE (CHARGING & GATEWAY) 52
23.5. TEXT TO SHOW IF VAT/SALES TAX IS 0% (CHARGING & GATEWAY) 52
23.6. 通貨用 YAML の設定 52
23.6.1. OpenShift での通貨設定の変更 53
23.6.2. 新しい通貨の確認 54
23.7. BILLING PERIODS FOR INVOICE IDS (CHARGING & GATEWAY) 54
23.8. CREDIT CARD POLICIES 55

第24章 支払い用のクレジットカードゲートウェイ **56**
24.1. 3SCALE でサポートされるクレジットカードゲートウェイ 56
24.2. クレジットカードゲートウェイとしての STRIPE の設定 56
24.2.1. 3scale 管理ポータルでの Billing API スコープを設定したアクセストークンの生成 57
24.2.2. Stripe からのキーおよび Webhook シークレットの取得 57
24.2.3. 3scale 管理ポータルでの課金の設定 58
24.2.4. 3scale デベロッパーポータルでのクレジットカード情報の編集 58
24.2.5. unsuccessfully charged 電子メール応答のテキストを更新する 58
24.3. BRAINTREE のクレジットカードゲートウェイとしての設定 59
24.4. デベロッパーポータルを介した拒否された請求書の支払いの許可 61
24.5. クレジットカードゲートウェイの問題のトラブルシューティング 61
24.5.1. 非推奨の支払いゲートウェイ 62
24.5.1.1. Adyen インテグレーション 62
24.5.1.1.1. 前提条件 62
24.5.1.1.2. Adyen インテグレーションの設定 63
24.5.1.2. Ogone インテグレーション 66
24.5.1.2.1. ステップ 1:Ogone からの API キーの取得 66
24.5.1.2.2. ステップ 2:3scale での設定 68
24.5.1.3. Authorize.Net インテグレーション 71

第25章 課金設定 **72**
25.1. 課金ルール 72
25.2. 課金ルールの設定 73
25.3. 既存課金ルールの更新 73

第26章 請求書 **74**
26.1. 請求書の一覧表示 74
26.2. 請求書ビュー 74
26.2.1. 請求書の詳細 74
26.2.2. 請求書の編集 75
26.3. 請求書のステータス 75
26.4. 自動請求プロセス 75
26.4.1. 各月の初日 75
26.4.2. 毎日 76

26.4.3. 自動および手動の請求書生成 76
26.4.4. 期中のアップグレード 77
26.4.4.1. 前払い請求モード 77
26.4.4.2. 後払い請求モード 78
26.5. アカウントごとの請求/課金の有効化/無効化 78
26.6. トライアル期間が設定されたプラン 79
26.7. VAT レート/消費税 79
26.7.1. VAT rate フィールドの設定 80
26.7.2. VAT code フィールドの設定 80
26.7.3. アカウントの VAT 値の設定 80
26.7.4. VAT が含まれる請求書 81
26.8. デベロッパーポータルビュー 81
26.9. クレジットカードに関する作業フロー 81
26.9.1. 有料プランへのサインアップ 81
26.9.2. 無料プランから有料プランへのアップグレード 82
26.10. 請求先住所フィールド 82

第27章 メール通知 84
27.1. プロバイダーへの通知 84
27.2. 開発者へのメール通知 84
27.2.1. 請求に関するメールアドレス 84

第28章 BILLING API 86

パート V. 解析 87

第29章 API アクセスを管理および最適化するための 3SCALE API 解析の実装 88
29.1. API の使用状況を把握する 3SCALE API メトリクスおよびメソッド 88
29.2. API メトリクスを取得するための 3SCALE プラグインの設定 89
29.3. 3SCALE API バックエンドの解析値の表示 89
29.4. API メトリクスを取得するためのプラグイン設定を確認するためのテストリクエストの送信 90
29.5. 3SCALE API の解析が表示されない場合のトラブルシューティングテクニック 91

第30章 組み込み機能を超える 3SCALE API 分析値のエクスポート 93
30.1. 3SCALE を使用してアプリケーションの使用状況に関するデータを抽出する例 93
30.2. カスタムの手順による 3SCALE アプリケーションの解析値の抽出 93

第31章 アプリケーションに対する 3SCALE の組み込みトラフィック解析の表示 97

第32章 API に対する 3SCALE レスポンスコードログの設定および評価 99

前書き

本ガイドは、管理ポータルで利用可能な機能を使用して 3scale インストール環境を管理する際に役立ちます。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、CTO である Chris Wright のメッセージ をご覧ください。

パート I. アカウント設定

第1章 アカウント設定

アカウントを作成したら、ご自分の会社に関する基本情報を更新します。住所を設定し、連絡先情報を追加します。

注記

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

1.1. 会社情報の追加

新規アカウントを作成したら、以下の手順により会社情報を追加します。

1. 上部ナビゲーションバーの右にある歯車アイコンをクリックします。**Overview** ウィンドウが表示されます。
2. **Account Details** の見出しの横にある **Edit** のリンクをクリックします。
3. アカウントの情報を入力します。

ここで指定する住所には、2 つの用途があります。

- 有料プランを利用している場合には、当社は請求のためにこの住所を使用します。
- 請求および支払いモジュールを使用している場合には、顧客に送付する請求書にもこの住所が表示されます。

1.2. 希望するタイムゾーンの選択

同じページで、すべてのシステム画面で使用されるタイムゾーンを選択することもできます。解析のグラフには、この設定が反映されます。ただし、請求サイクルの計算は UTC 時間に従って処理されます。

第2章 3SCALE 管理ポータル用 RED HAT SINGLE SIGN ON

本章では、3scale 管理ポータルで Red Hat Single Sign On (RH-SSO) を設定して使用する方法について説明します。

2.1. RH-SSO または AUTH0 メンバー認証の有効化

3scale では、メンバーおよび管理者に対するシングルサインオン (SSO) 認証がサポートされます。

3scale 管理ポータルでは以下の SSO プロバイダーがサポートされ、それぞれさまざまな ID ブローカリングおよびメンバーフェデレーションオプションに対応しています。

- Red Hat SSO (RH-SSO)
- Auth0

注記

複数の SSO メンバー認証タイプを有効にすることができます。

RH-SSO または Auth0 に追加されているユーザーだけが、SSO を通じて 3scale 管理ポータルにアクセスすることができます。ロールまたはユーザーグループによりさらにアクセスを制限する場合には、RH-SSO または Auth0 のサポートポータルで、該当するステップバイステップのチュートリアルを参照してください。

選択したプロバイダーで SSO を確立したら、それを 3scale 管理ポータルで設定して有効にする必要があります。

2.1.1. RH-SSO 使用の前提条件

- Creating the Developer Portal の Developer Portal authentication の章に記載の手順に従って、RH-SSO インスタンスおよびレルムが設定されている。
- 3scale のエンタープライズアカウント

2.1.2. Auth0 使用の前提条件

- Auth0 のサブスクリプションおよびアカウント

2.1.3. RH-SSO の有効化

RH-SSO または Auth0 を有効にするには、3scale 管理ポータルで管理者として以下の手順を実施します。

1. 希望の SSO プロバイダー (前提条件を参照) が正しく設定されていることを確認します。
2. Account Settings の SSO Integrations に移動します。
   - ページ右上隅にある歯車アイコンをクリックします。
   - **Account Settings (歯車アイコン) > Users > SSO Integrations**の順に移動し、**New SSO Integration** をクリックします。
3. ドロップダウンリストから、希望の SSO プロバイダーを選択します。

4. 必須の情報を入力します (SSO を設定した際に指定したもの)。
    - Client
    - Client Secret
    - Realm or Site
5. **Create Authentication Provider**をクリックします。

注記

テスト中にコールバック URL の不一致が発生した場合には、エラーメッセージに示されるコールバック URL を Auth0 が許可するコールバック URL に追加します。

2.2. 3SCALE での RH-SSO の使用

SSO を設定したら、メンバーは接続された IdP のアカウントクレデンシャルを使用してサインオンすることができます。

SSO を使用して 3scale 管理ポータルにログインするには、以下の手順に従います。

1. 3scale のログインページに移動します。

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

2. ご自分の IdP で 3scale を承認します。
3. 状況に応じて、必要な情報を入力してサインアップを完了します。

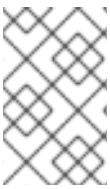
サインアップに成功すると、API プロバイダー組織下にメンバーアカウントが作成され、自動的にログインされます。

2.3. 3SCALE ログインの RH-SSO オプションへのリダイレクト

本セクションでは、RH-SSO を使用したアイデンティティープロバイダー (IdP) のログインウィンドウへのリダイレクトについて説明します。3scale API Management の管理者は、以下の手順を実施して、ご自分の 3scale アカウントをオプションのシングルサインオン (SSO) のログインページ経由でアクセス可能にすることができます。

2.3.1. 前提条件

- 3scale 2-saas
- Creating the Developer Portal の Configuring RH-SSO セクションに記載の手順に従って、RH-SSO インスタンスおよびレルムが設定されている。



注記

RH-SSO を 3scale と統合するためには、動作状態にある RH-SSO インスタンスが必要です。インストール手順については、RH-SSO のドキュメントを参照してください。RH-SSO 7.2 のインストール

2.3.2. 必要な手順

1. 3scale ドキュメントの Red Hat Single Sign-On for the 3scale Admin Portal セクションの RH-SSO を設定するための手順にアクセスし、それに従ってください。

2. RH-SSO の管理者に 3scale の URL を提供します。この URL が RH-SSO 内でリダイレクトの基盤となり、セキュアなログオンが確保されます。URL には以下のフォーマットを使用します。

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

3. **<system_name>** は、管理ポータルの SSO Integration の詳細ページで取得することができます。

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

4. **Keycloak_0123456aaaaa** も、以下のように SSO Integration の詳細ページの **Callback URL for OAuth flow test** フィールドから確認することができます。

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

第3章 ユーザーの招待および権限の管理

注記

招待 機能を利用することができるのは、Pro および Enterprise のお客様だけです。

API 管理のワークロードを分散するために、3scale 管理ポータルにアクセスできるように同じ組織のチームメンバーを招待したい場合があります。以下の手順で、1 人または複数のチームメンバーに 3scale 管理ポータルへのアクセス権限を付与する方法を説明します。

この章で言う **ユーザー** とは、ご自分のチームのメンバーを指します。3scale 管理ポータルには、2 種類のユーザーが存在します。

- **管理者:**すべての機能とサービスに対するフルアクセス権限を持ち、他のメンバーを招待することができる (プランで許可される場合)。
- **メンバー**一部の製品機能に対するアクセス権限を持ち (例: 解析、デベロッパーポータル)、Enterprise のお客様であればサービスに対しても限定的なアクセス権限を持つ。

シングルサインオン (SSO) インテグレーションから新しい 3scale ユーザーを作成する場合、SSO トークンの内容に関係なく、このユーザーはデフォルトで **メンバー** ロールを持ちます。3scale では、そのロールを SSO ロールにマッピングしません。

3.1. ユーザー管理画面への移動

ご自分の 3scale インストール環境内のユーザーを一覧表示するには、管理ポータルのページで以下の手順に従います。

1. ウィンドウ上部のナビゲーションバーの右にある歯車アイコンをクリックします。
2. 左側のメニューから、**Users > Listing**の順に移動します。

3.2. 招待状の送信

ユーザーのリストから、新しいチームメンバーを招待することができます。招待状を送信するには、以下の手順を実施します。

1. リストの右上にある **Invite user** のリンクをクリックします。
2. 招待するユーザーのメールアドレスを入力し、**Send** をクリックします。
3. 送信の確認として、ウィンドウの右上隅に以下のメッセージが表示されます。**Invitation was successfully sent.**.

入力したアドレスに招待メールが送信されます。電子メールが届かない場合には、受取人のメールアカウントで迷惑メールと識別されていないことを確認します。

また、**Users > Invitations**で、送信した招待状のリストおよびステータスを確認することができます。

3.3. 招待の応諾

新たな管理者またはメンバーは、招待メールのリンクをクリックし、フォームに入力してプロセスを完了する必要があります。フォームが送信されると、アカウントがアクティブになります。

3.4. 新しいユーザー権限の付与

チームメンバーへの権限付与には、大きく分けて 2 つのタイプがあります。

- **機能別**:解析、請求、開発者の管理など。
- **サービス別**:すべてのサービスの中から、メンバーにアクセス権限を付与するサービスを選択します。**注記**:この機能を利用することができるのは、Enterprise のお客様だけです。

新しいユーザー権限を付与するには、ユーザーメニューからユーザーを選択し、**Edit** をクリックしてそのユーザーを編集します。ユーザーロールには、以下のタイプがあります。

- ユーザーのロールを **Admin** に変更すると、管理ポータルを制御するためのフルアクセス権限が付与されます。
- ユーザーのロールを **Member** に変更すると、さらにチームメンバーがアクセス権限を持つ機能およびサービスを選択することができます。
  - **Member** の場合には、機能を選択すると、その機能に関して利用可能なすべてのサービスが一覧表示されます。

管理ポータルの特定機能へのアクセス権限を付与すると、メンバーには該当する API へのアクセス権限だけが与えられます。

- 開発者アカウント – アプリケーション:Account Management API へのアクセス権限を付与します。
- 解析:Analytics API へのアクセス権限を付与します。
- 請求:Billing API へのアクセス権限を付与します。

第4章 通知

Red Hat 3scale API Management のユーザーインターフェイスおよび 3scale API に関する操作を行うと、通知が送信されます。開発者がトリガーとなり、管理者、その他のメンバー、および 3scale に通知が送信されます。API 通知については、アクセス権限のある API に関する通知しか受け取ることができません。

4.1. 通知のタイプ

以下に示すように、通知にはさまざまなタイプがあります。

- アカウント
- 請求
- アプリケーション
- サービスサブスクリプション
- 使用量に関するアラート

4.2. 制約

管理ユーザーは、すべての通知に対するアクセス権限を持ちます。

メンバーユーザーは、アクセス権限が付与されている機能に関する通知にしかアクセスすることができません。たとえば、メンバーに請求機能へのアクセス権限が付与されている場合には、そのメンバーは請求に関する通知にのみアクセスすることができます。

Enterprise アカウント の場合には、メンバーユーザーは、アクセス権限が付与されているサービスの活動に関する通知にしかアクセスすることができません。

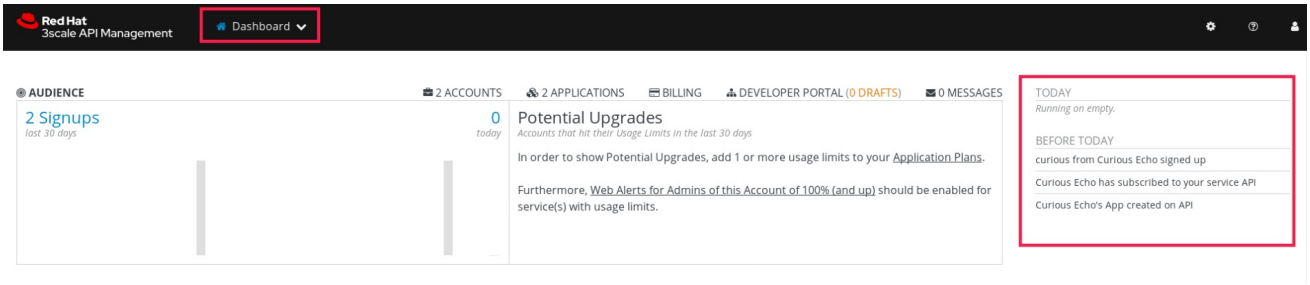
4.3. メール通知へのサブスクライブ

サブスクリプションは個人的な設定で、これらの通知を受け取るユーザーだけが変更可能です。ご自分のサブスクリプションを編集するには、以下の手順を実施します。

1. **Account Settings (ナビゲーションバーの歯車アイコン) > Personal > Notification Preferences** の順に移動します。
2. 受け取りを希望する通知を選択します。
3. **Update Notification Preferences** をクリックします。

4.4. WEB 通知

メール通知に加えて、Dashboard で活動に関する履歴情報を確認することができます。


Red Hat
3scale API Management
Dashboard
AUDIENCE
2 ACCOUNTS
2 APPLICATIONS
BILLING
DEVELOPER PORTAL (0 DRAFTS)
0 MESSAGES
2 Signups
last 30 days
0
today
Potential Upgrades
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.
TODAY
Running on empty.
BEFORE TODAY
curious from Curious Echo signed up
Curious Echo has subscribed to your service API
Curious Echo's App created on API

第5章 個人設定

Personal 設定では、チームメンバーとして個人設定を編集することができます。管理者の場合には、アカウント設定を編集することもできます。そのためには、アカウント設定 のチュートリアルを参照してください。

個人設定を編集するには、以下の手順を実施します。

1. ナビゲーションバーの歯車アイコンをクリックします。
2. 左側のパネルで **Personal** に移動します。ここから 3 つのタイプの設定を編集することができます。
    - **個人情報**:名前、電子メール、パスワードなど
    - **トークン**:3scale API (Billing、Account Management、および Analytics) に対する認証を行うためのアクセストークンを作成し、ActiveDocs (インタラクティブドキュメント) 内から使用します。詳細は、3scale トークン を参照してください。
    - **通知設定**:受け取りを希望する通知を選択します。注記:Enterprise のお客様でメンバーの場合には、表示される通知は機能およびサービスでフィルターリングされます。つまり、アクセス権限が付与されている機能およびサービスに関する通知だけをサブスクライブすることができます。通知の個人設定に関する詳細は、通知 を参照してください。

第6章 トークン

本チュートリアルでは、トークンとは何か、その仕組み、作成方法など、3scale トークンについて説明します。

3scale には 2 種類のトークンがあります。**アクセストークン** (ユーザーが作成) および **サービストークン** (3scale で新規サービスを作成する際に自動的に作成)。

6.1. アクセストークン

アクセストークンにより、API プロバイダーの管理者およびメンバーは、3scale API (Billing、Account Management、および Analytics) に対する認証を行うことができます。トークンは、ActiveDocs (インタラクティブドキュメント) 内から使用します。

アクセストークンにより、読み取り/書き込みアクセス権限、または読み取り専用アクセス権限のいずれかが許可されます。

考慮すべき重要なことは、メンバーの権限に応じてアクセストークンの機能する仕組みが変わるという点です。管理者は、3 つすべての 3scale API に対する認証を行うためのトークンを作成することができます。メンバーの場合には、アクセス権限を持つ管理ポータル機能によって作成できるトークンが制限されます。たとえば、請求機能に対するアクセス権限を持たないメンバーは、Billing API に対する認証を行うためのトークンを作成することができません。

6.2. アクセストークンの作成

アクセストークンは、Tokens のページで作成することができます。トークンを作成するには、以下の手順に従います。

1. ナビゲーションバーの歯車アイコンをクリックします。
2. **Personal > Tokens**の順に移動します。
3. **Add Access Token** をクリックします。
4. 名前を指定し、対象となる API を 1 つまたは複数選択し、トークンの権限を選択します。
5. 新しいトークンを保存するには、**Create Access token** をクリックします。

メンバーの場合には、すべての API が表示されるとは限らない点に注意してください。アカウントの管理者からアクセス権限が付与された API だけが表示されます。

アクセストークンは、必要なだけ作成することができます。セキュリティー上の理由から、トークンは 3scale に保存されません。新しいトークンを作成すると、トークンを保存するように要求されます。保存したそのトークンを使用して、3scale API にリクエストを行うことができます。

トークンを紛失した場合には、トークンを削除することを推奨します。これにより、トークンが無効になり、効力が無くなります。その後、新しいトークンを作成します。

6.3. アクセストークンの使用

アクセストークンを使用して 3scale API を呼び出すと、アクセス権限のあるサービスによって結果が絞りこまれます。

たとえば、Self-managed APIcast をデプロイする場合、APIcast API ゲートウェイが Account Management API を使用してサービスの設定をプルできるように、アクセストークンが必要になります。

アクセストークンが機能する仕組みは、以下のとおりです。ご自分の組織が 3scale で 3 つのサービスを設定していると仮定します。メンバーであるあなたは、サービス 1 へのアクセス権限はあるが 2 および 3 へのアクセス権限はありません。また Account Management API へのアクセス権限があります。この場合には、トークンを作成して Account Management API にリクエストを行うと、サービス 1 を使用するアプリケーションだけを取得することができます。

同じ例において、Account Management API へのアクセス権限はあるがどのサービスへのアクセス権限も持たない場合には、呼び出しを行うと access denied エラーが返されます。

6.3.1. サービストークン

サービストークンを使用して、3scale Service Management API に対する認証を行います。サービストークンは、3scale で新規サービスを作成する際に自動的に生成され、サービスごとに固有です。このトークンは、3scale アカウントのユーザー間で共有されます。

管理ポータルの Dashboard で、ユーザーがアクセス可能なサービスのサービストークンを確認することができます。**Account Settings (ギアアイコン) > Personal > Tokens**

パート II. 開発者アカウントの管理

第7章 開発者の追加

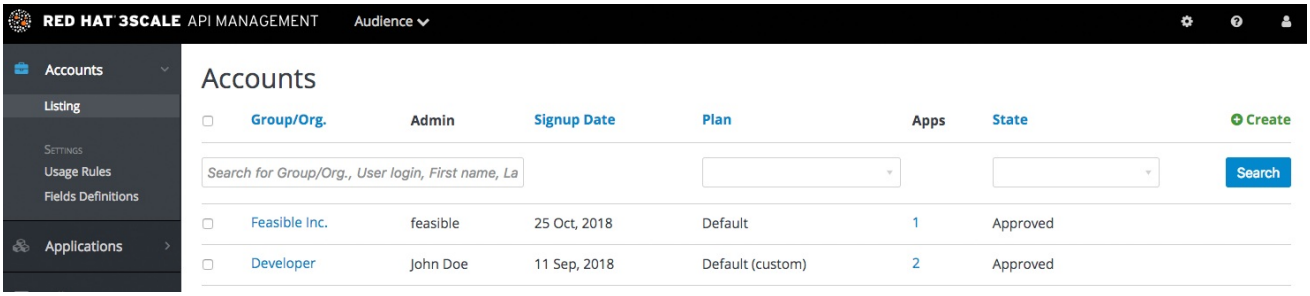
API へのアクセス用に新たな開発者アカウントを追加する手順を以下に示します。

手動で開発者を招待するワークフローを設定している場合に、ここでは新たな開発者を追加する方法について説明します。

7.1. 新たな開発者アカウントの作成

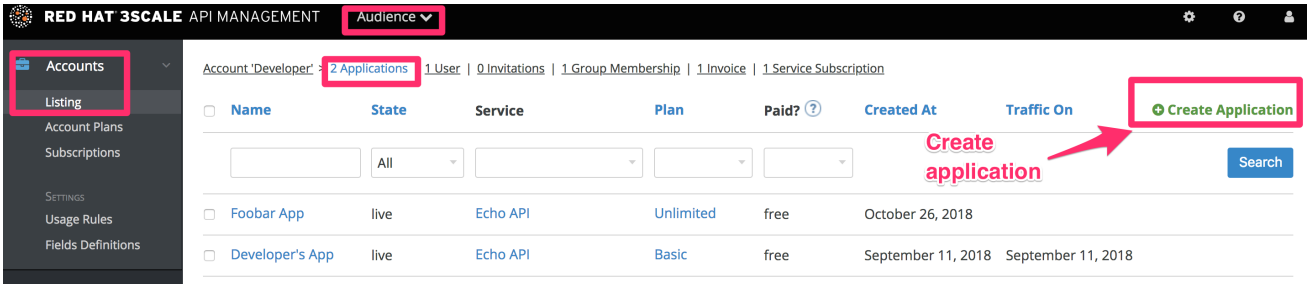
1. 管理ポータルの **Audience** セクションで、**Accounts** のリンクをクリックします。
2. **Create** をクリックします。

管理者であれば、必須フィールドの一部を省略することができます。ユーザーを安全にアカウントに招待する場合には、パスワードフィールドも省略することができます。ただし、このメインの管理アカウントのメールアドレスは、すべてのユーザーに対して一意でなければなりません。



7.2. アプリケーションの設定

アカウントのアプリケーションキーを事前に設定する場合には、開発者に代わってアプリケーションを追加することもできます。そうでなければ、開発者が行う初期ステップの一部として、本セクションを省略します。



7.3. 開発者への通知

手動で開発者に招待メールを送信するか、手順に従って 開発者の招待 機能を使用することができます。

第8章 開発者の承認

本セクションでは、サインアップワークフローの各ステップで承認を行う方法について説明します。

手動の承認ステップを含むサインアップワークフローを実装したら、いくつかのオプションを選択することができます。承認プロセスには、そのトリガーおよび承認の対象によって若干の違いがあります。メール通知を受け取った場合には、次のセクションの指示に従ってください。それ以外の場合には、承認の対象 (アカウント、サービス、またはアプリケーション) より手順が異なります。

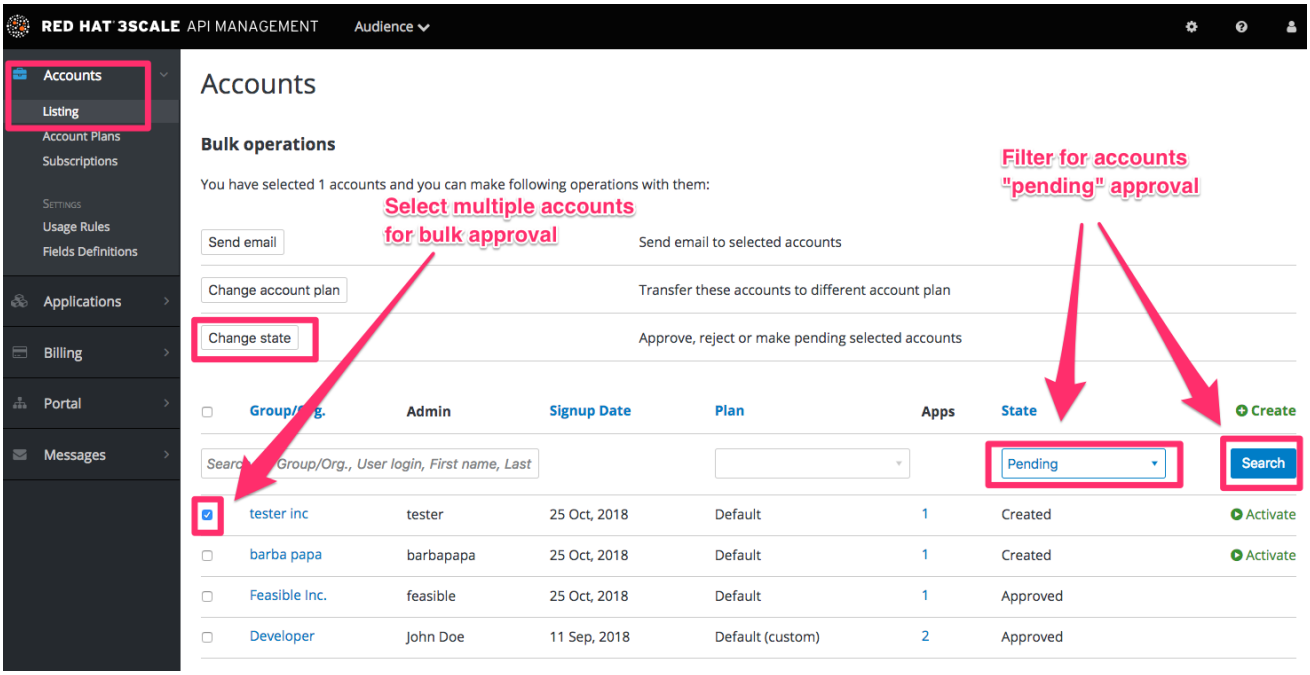
8.1. メール通知からの承認

管理者が電子メールを受け取り、開発者の 1 人に承認待ちの項目があることを認識した場合には、その承認項目の URL をコピーしてブラウザーに貼り付けると、承認を行うページに直接移動することができます。

8.2. アカウントの承認

特定のアカウントを検索する、または (承認が)pending ステータスにあるアカウントに絞り込むには、**Audience** > **Accounts** > **Listing**の順に移動します。承認待ちのアカウントだけを表示するには、**State** のドロップダウンリストで Pending を選択し、**Search** をクリックします。

それぞれの行で個別に直接承認を行うことも、一度に複数の行を選択して一括して承認を行うこともできます。

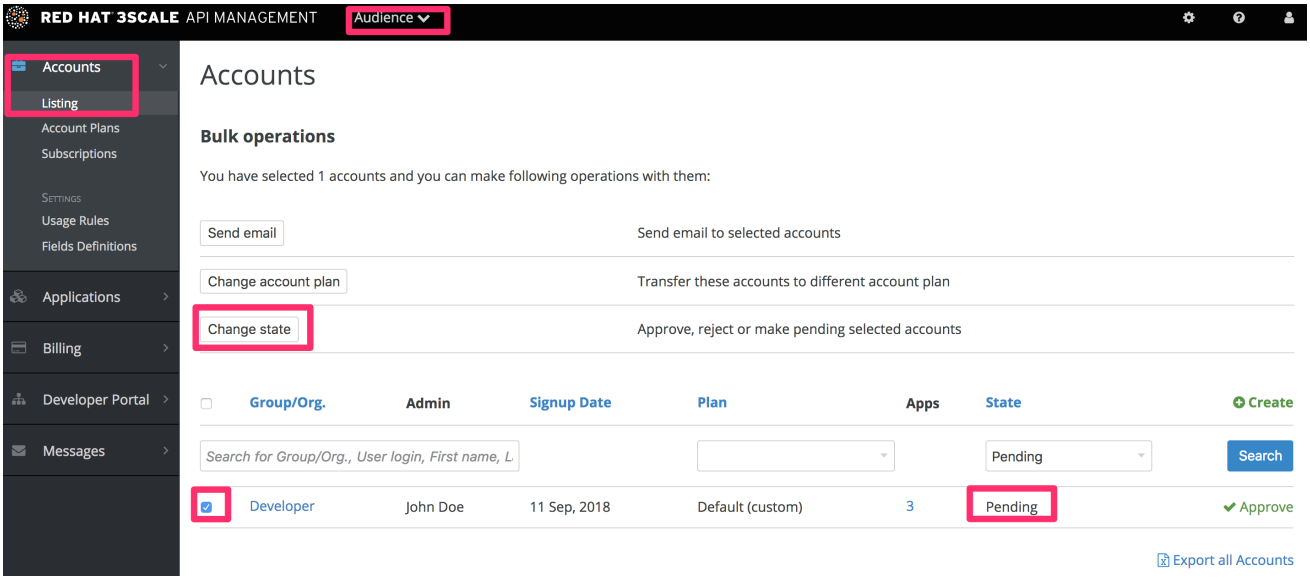


8.3. サービスの承認

特定のサービスサブスクリプションを検索する、または (承認が)pending ステータスにあるサブスクリプションに絞り込むには、**Audience** > **Accounts** > **Subscriptions**の順に移動します。

サブスクリプションを表示するには、**Audience** > **Accounts** > **Usage Rules**で Service Plans のチェックボックスを選択します。

1 つのサブスクリプションを選択することも、一度に複数のサブスクリプションを選択して一括して承認を行うこともできます。

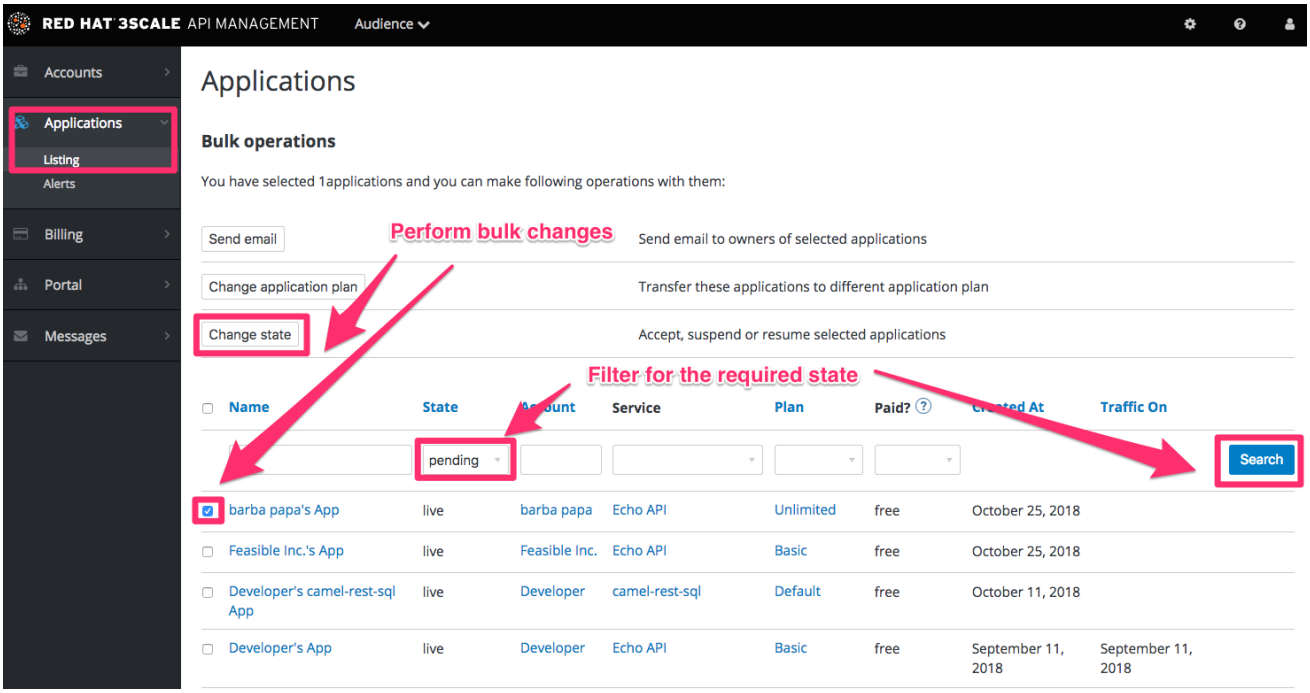


8.4. アプリケーションの承認

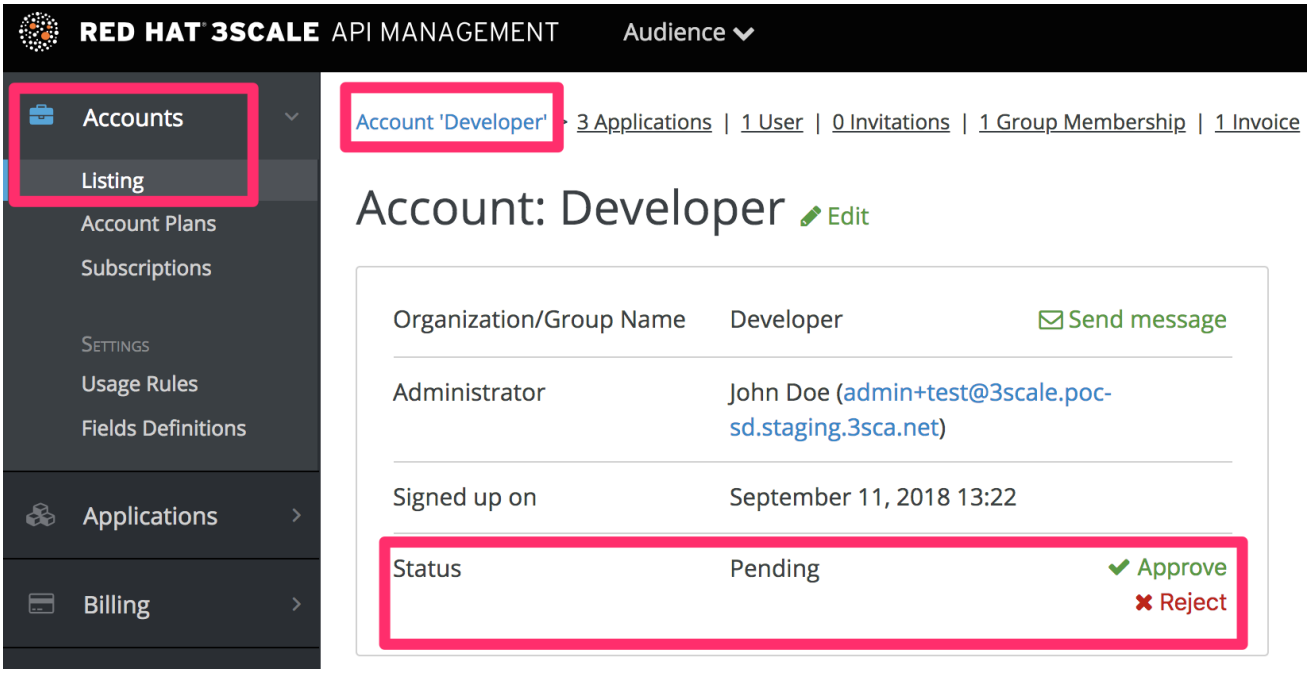
特定のアプリケーションを検索する、または (承認が)pending ステータスにあるアプリケーションに絞り込むには、以下の手順を実施します。

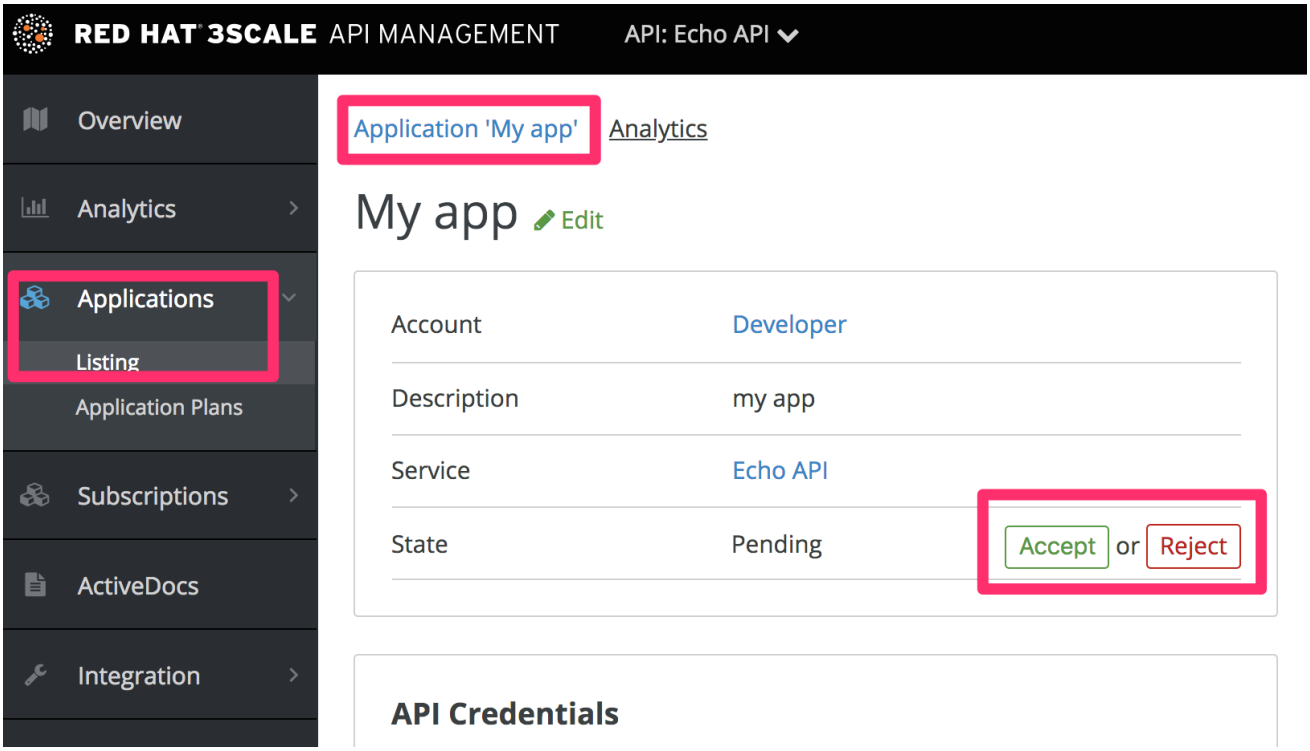
1. **Audience** > **Applications** > **Listing**の順に移動します。
2. **State** のドロップダウンリストで Pending を選択し、 **Search** をクリックします。

1 つのアプリケーションを選択することも、一度に複数のアプリケーションを選択して一括して承認を行うこともできます。



開発者アカウントの詳細ページから操作を開始することもできます。そこで承認するアプリケーションを選択し、アプリケーションの詳細ページで承認を行います。


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
Send message
Administrator
John Doe (admin+test@3scale.poc-sd.staging.3sca.net)
Signed up on
September 11, 2018 13:22
Status
Pending
Approve
Reject


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
Accept
or
Reject
API Credentials

第9章 アプリケーションプランの変更

本セクションでは、アカウント、サービス、またはアプリケーションのプランを変更する方法について説明します。

管理者は、自分の判断で、または開発者からのプラン変更要求に対応して、開発者のプランを変更する場合があります。

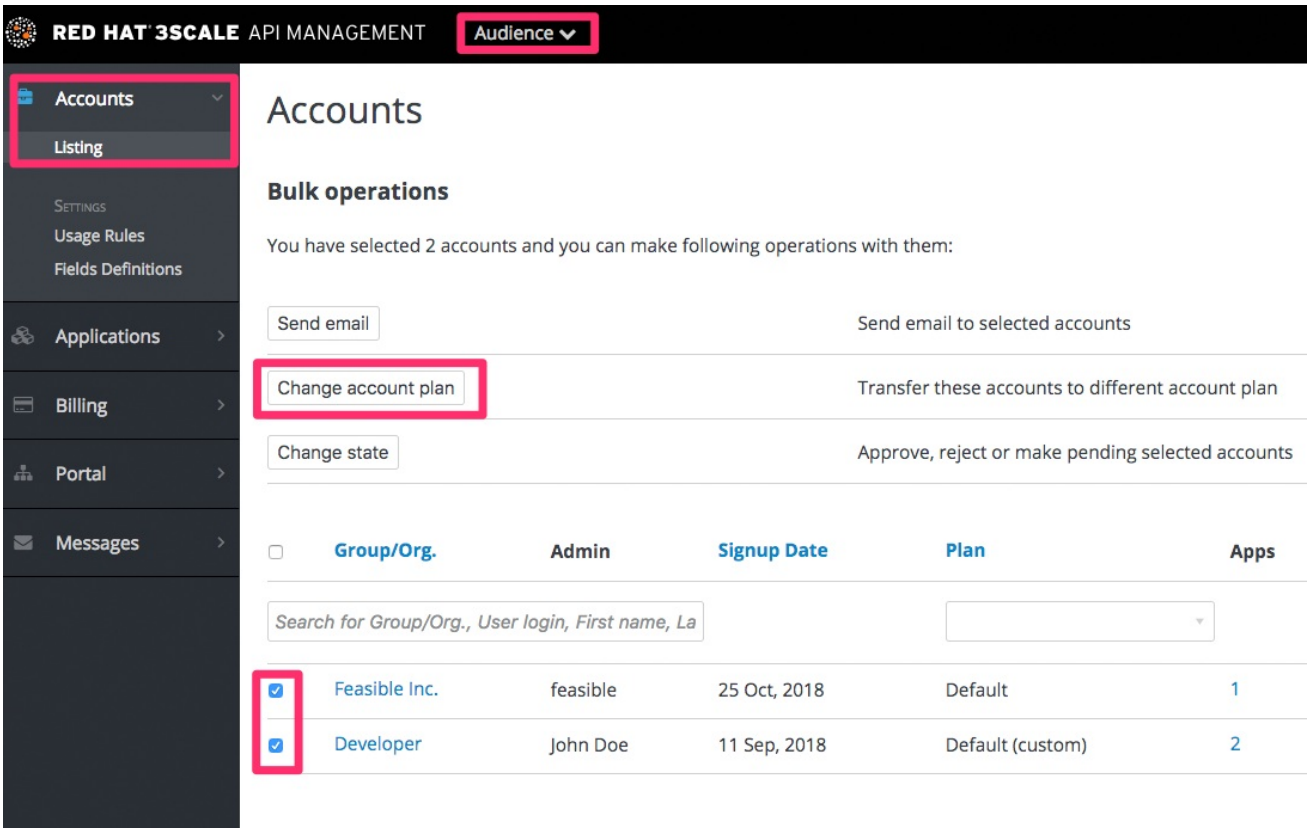
注記

プラン変更の手順には、変更するプランのタイプごとに若干の違いがあります。

9.1. アカウントプランの変更

特定のアカウントを検索する、または特定のアカウントに絞り込むには、**Audience > Accounts > Listing** の順に移動します。

1つまたは一度に複数の行を選択し、プランを変更することができます。

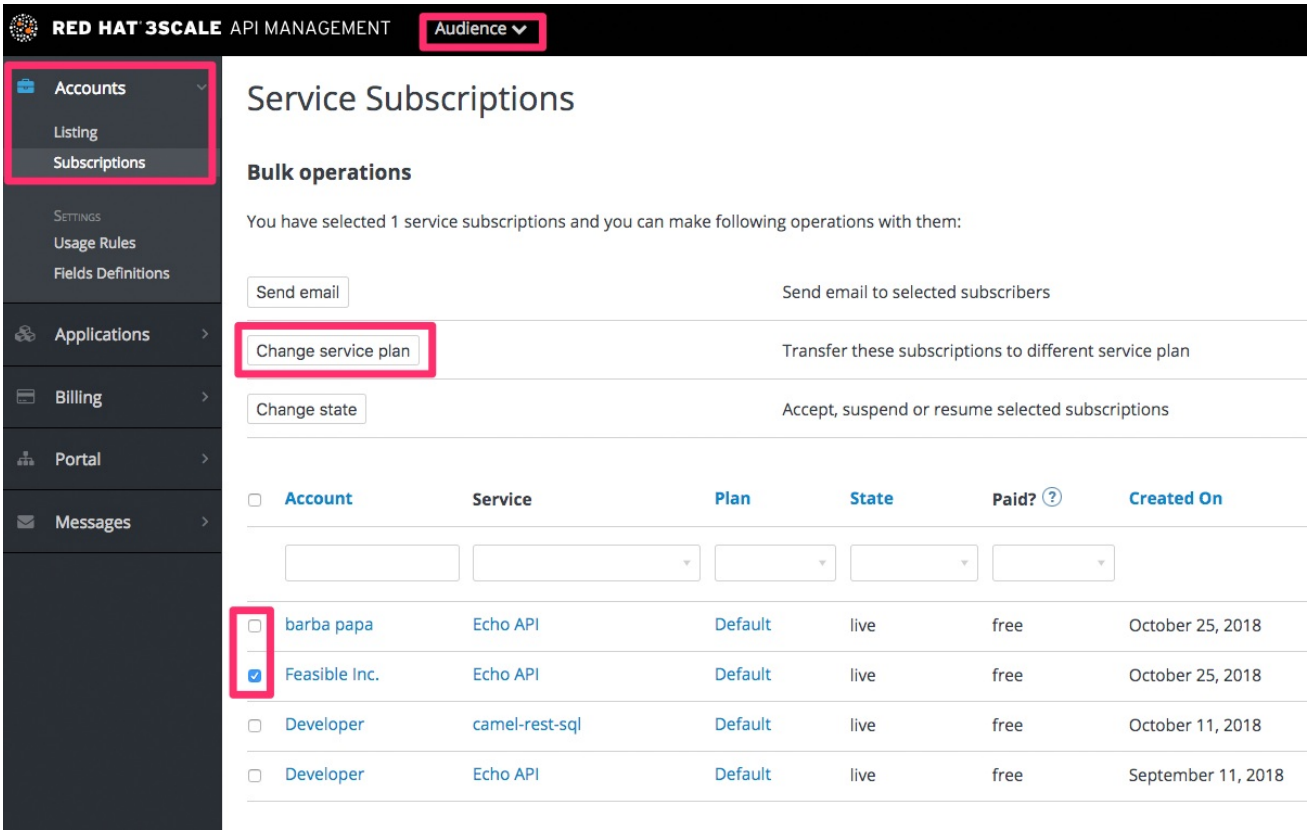


9.2. サービスプランの変更

特定のサービスサブスクリプションを検索する、または特定のサービスサブスクリプションに絞り込むには、**Audience > Accounts > Subscriptions**の順に移動します。

Settings のページで Service Plans のチェックボックスを選択している場合に限り、サブスクリプションが表示されます。

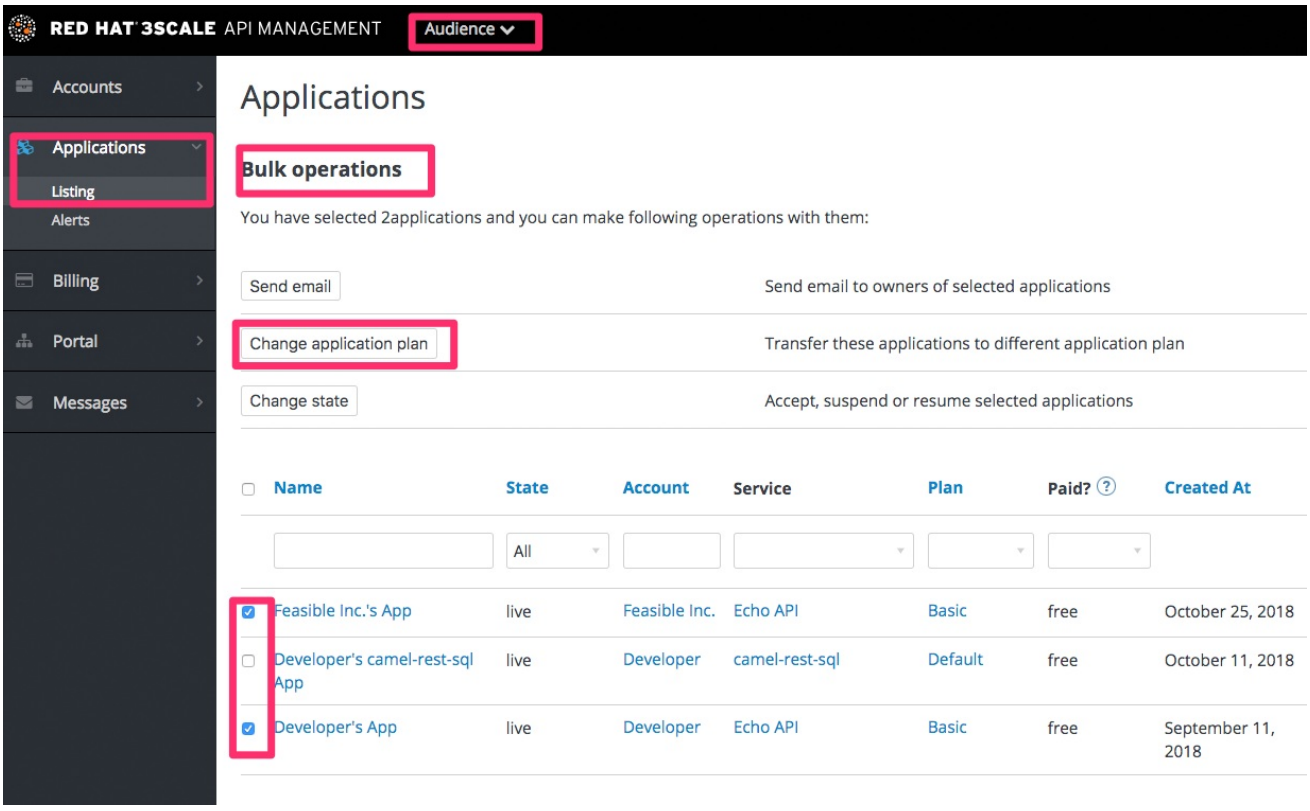
1つまたは一度に複数のサブスクリプションを選択し、プランを変更することができます。



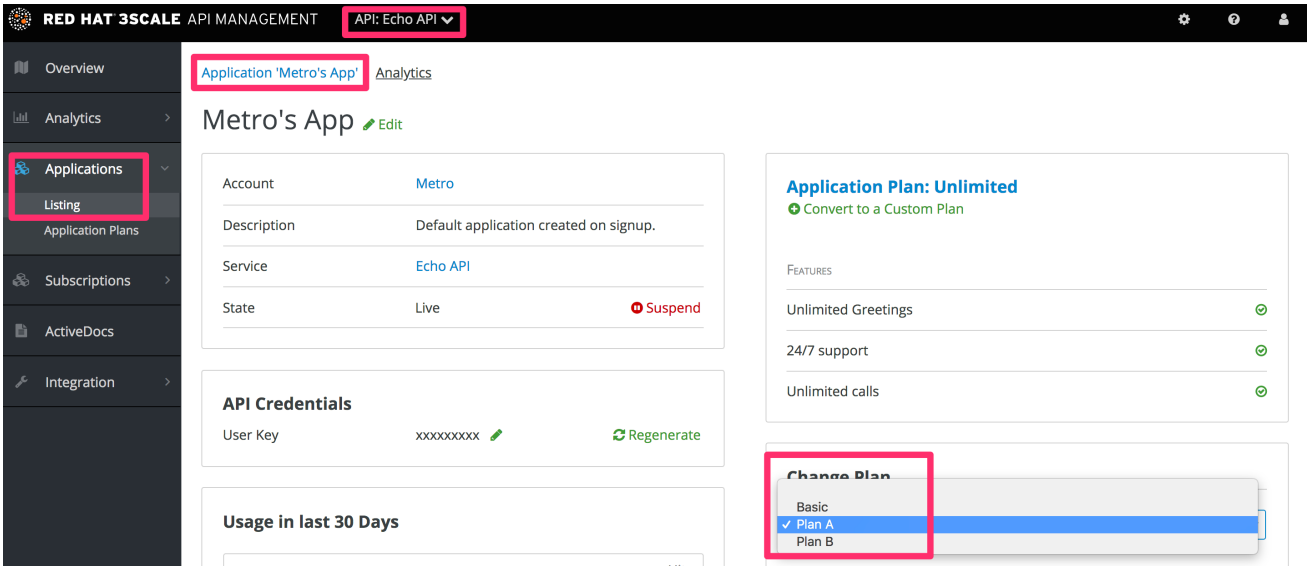
9.3. アプリケーションプランの変更

特定のアプリケーションを検索する、または特定のアプリケーションに絞り込むには、**Audience** > **Applications** > **Listing** の順に移動します。

1つまたは一度に複数のアプリケーションを選択し、プランを変更することができます。



もう1つのシナリオは、開発者アカウントの詳細ページから操作を開始するものです。そこでプランを変更するアプリケーションを選択します。アプリケーションの詳細ページで、プランを変更することができます。



9.3.1. 補足情報

別の標準プランに変更するのではなく、1つの特定アプリケーション用のプランに変更を加えるだけの場合には、プランのカスタマイズ 機能を使用することができます。

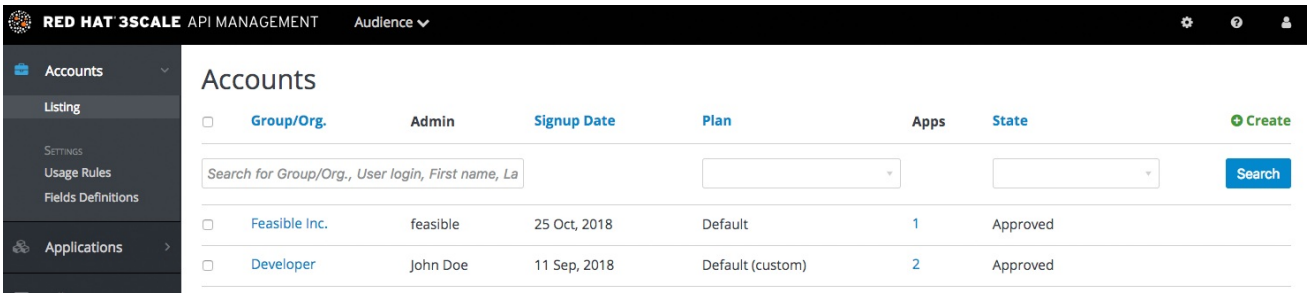
第10章 開発者への連絡

本章では、特定のアプリケーションを管理する開発者アカウントを把握して、3scale 機能および外部の手段を使用して開発者と連絡を取る方法について説明します。

API の運用中に、API を使用している開発者に緊急に連絡を取らなければならない場合があります。

10.1. システム内の該当アプリケーションおよびアカウントの把握

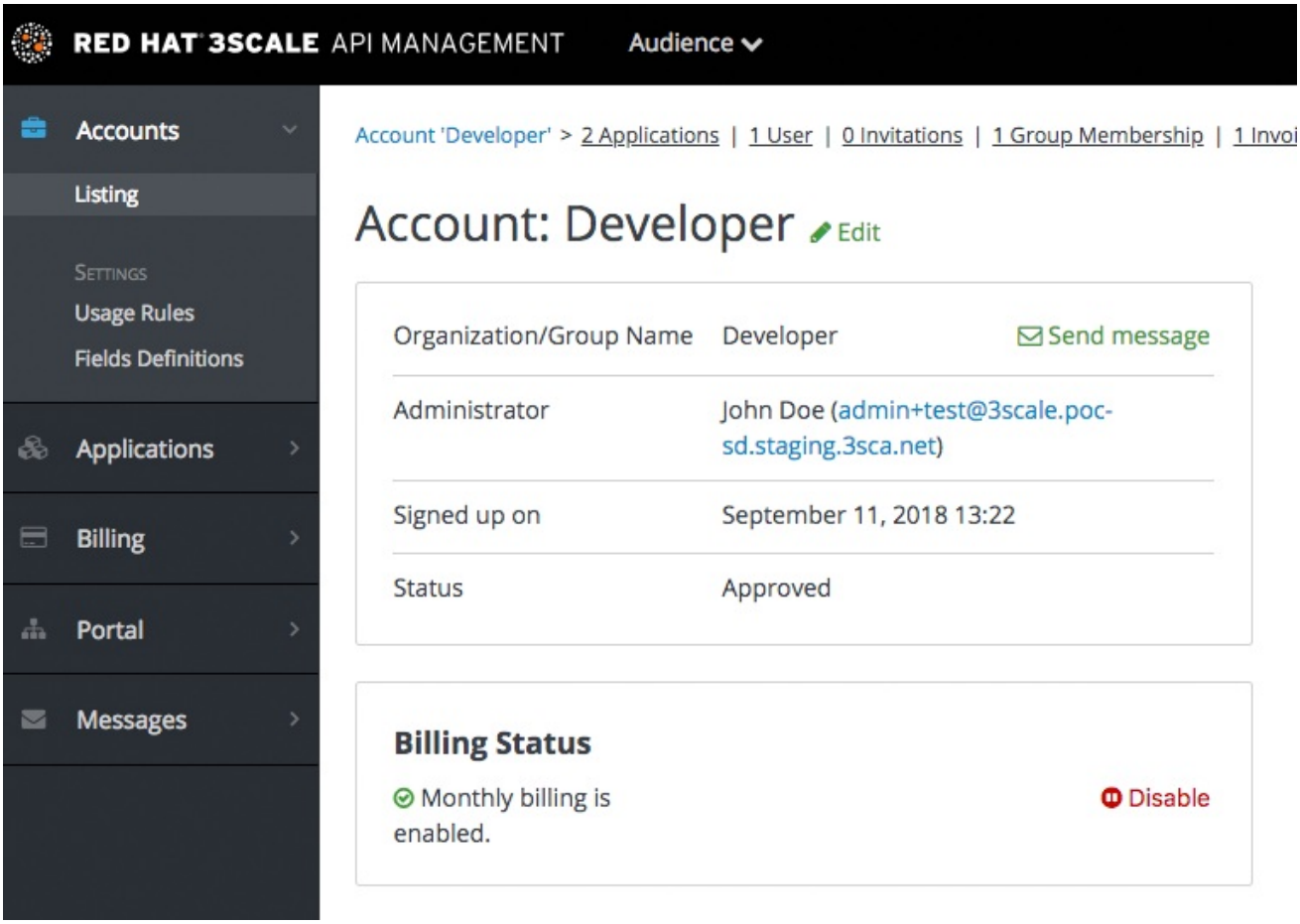
問題のアプリケーションを管理しているアカウントおよび開発者がすでに分かっている場合には、以下に示すように Accounts のページ (**Audience > Accounts > Listing**) からそのアカウントに移動します。



アプリケーション ID または API キーしか分からない場合には、Accounts のページ (**Audience > Accounts > Listing**) の検索ボックスを使用して、該当するアカウントを特定することができます。アプリケーションの特定に関する詳細は、ここ を参照してください。

10.2. 開発者への内部メッセージの送信

以下に示すように、アカウントプロファイルのページに移動したら、メッセージアイコンをクリックします。



ここで作成されるメッセージは、そのアカウントのすべての開発者がメッセージを確認できるアカウントシステムダッシュボードに送信されると共に、そのアカウントで管理者ステータスを持つ開発者アカウントのユーザーに電子メールで送信されます。

10.3. 他の手段による連絡

緊急事態が発生し電子メールの対応では間に合わないと考えられる場合には、サインアップ時に開発者が入力した連絡先情報を使用することもできます。この情報は、以下の場所に表示されます。

- 企業アカウントページ (一般的な連絡先情報が主ですが、電話番号が含まれる場合があります)
- ユーザー独自ファイルの開発者/ユーザー固有情報

サインアップ時に、連絡先電話番号を必須フィールドにできる点に注意してください。

第11章 プランのカスタマイズ

本セクションでは、特定の開発者向けにアプリケーションプランをカスタマイズする方法について説明します。

アプリケーションプランは、開発者コミュニティーのさまざまなグループに標準ポリシーを適用するのに適しています。ただし、独自の要件を持つ個々の開発者向けに、標準プランを柔軟にカスタマイズすることができます。

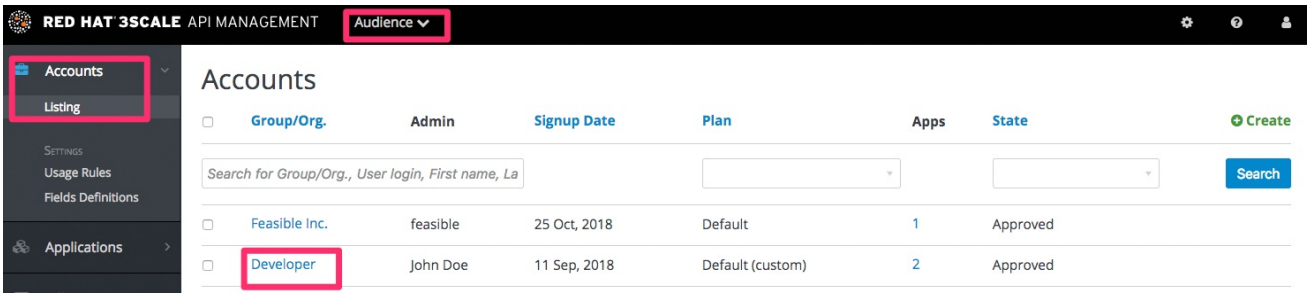
プランをカスタマイズすると、元のプランへのリンクは失われます。元のプランに変更を加えても、カスタムプランではこれらの変更は継承されません。したがって、カスタムプランが多くなりすぎて効果的に管理できなくならないように、このカスタマイズ機能は慎重に使用する必要があります。

開発者は、現在の請求期間がすでに始まっているので、次の価格帯にアップグレードせずに現在の制限を引き上げることを希望します。制限を引き上げ発生した変動費だけを請求するカスタマイズは、この状況に対処するのに適した手段です。このカスタマイズは、次の請求月にアップグレードを奨励するのにも役立ちます。

11.1. アカウントの選択

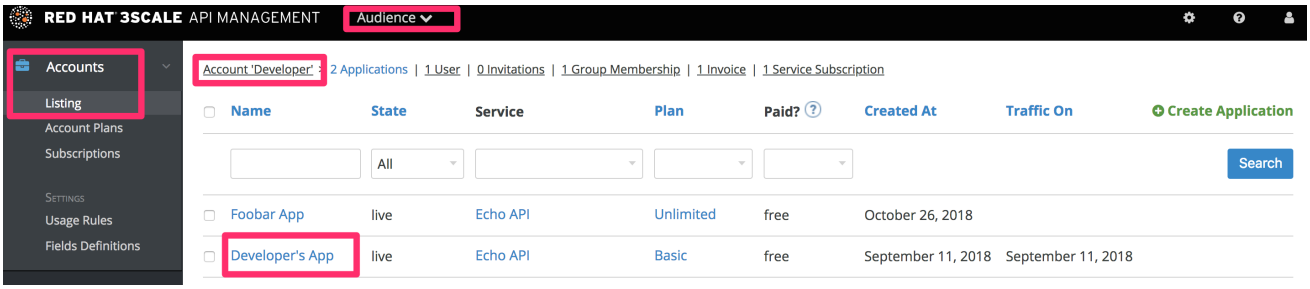
対象となる開発者アカウントの詳細ページを表示するには、以下の手順を実施します。

1. **Audience > Accounts > Listing**の順に移動します。
2. 開発者アカウントを選択します。



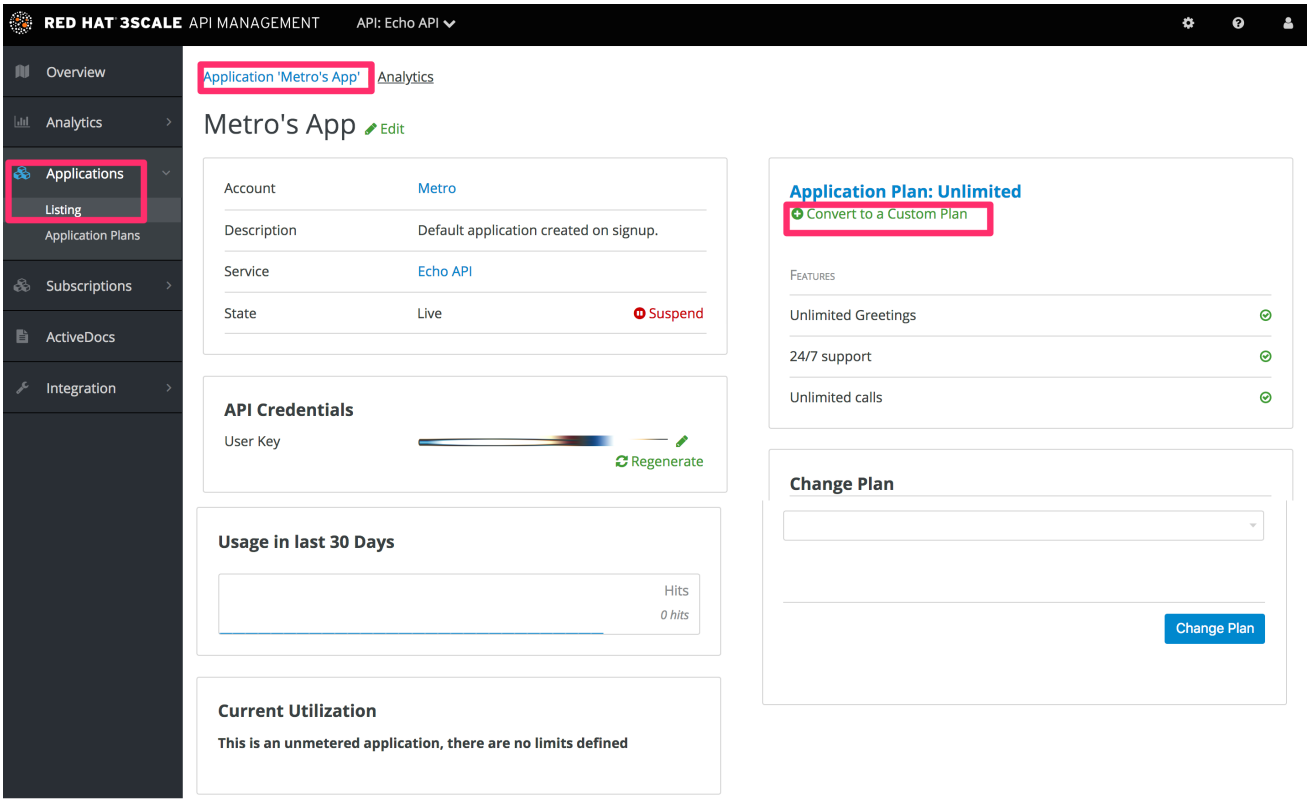
11.2. アプリケーションの選択

プランをカスタマイズするアプリケーションを選択します。



11.3. アプリケーションプランのカスタマイズ

カスタマイズするオプションを選択します。この操作により表示されるページで、このアカウントが所有するアプリケーションに適用されるプランの要素をすべてカスタマイズすることができます。



11.3.1. 補足情報

プランのカスタマイズを開始する前に、必ず初めに新しい標準プラン (デベロッパーポータルで非表示にすることができる) の方が良いかどうかを検討してください。標準プランが望ましければ、標準プランに変更 します。そのプランが複数の開発者パートナーに適用される場合には、再利用のメリットを得ることができます。

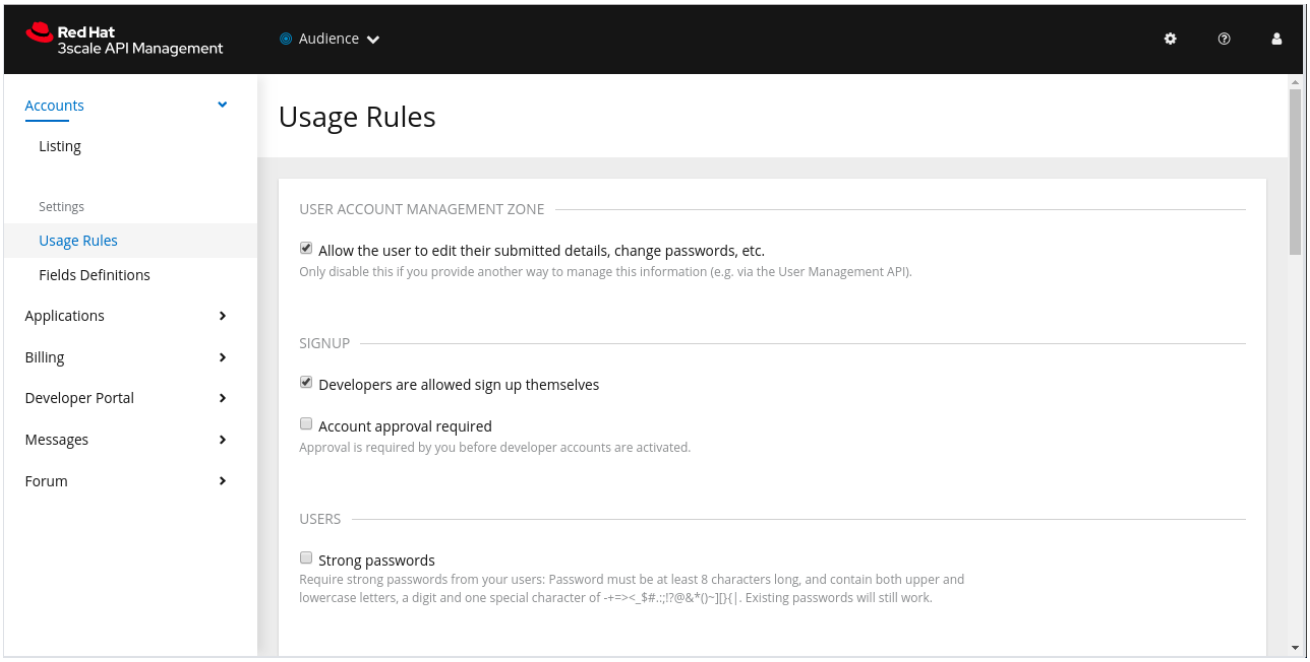
第12章 セルフサービスによるサインアップの有効化

セルフサービスモードまたは手動モードを実装して、開発者のサインアップを設定します。

開発者のサインアップワークフローを、セルフサービスまたは手動の招待のみのいずれかに設定することができます。セルフサービスのサインアップは、開発者がデベロッパーポータルを通じて行います。一方、手動の招待は管理者が管理ポータルを通じて処理します。

デフォルトでは、開発者は自分でサインアップを行うことができます。開発者のセルフサービスを有効にした場合には、開発者アカウントをアクティブにするのに、管理者承認を要求することができます。

そのためには、**Audience > Accounts > Settings > Usage Rules**の順に移動します。



第13章 アプリケーションの特定

本章では、名前、API キー、またはアプリケーションの ID のいずれかを元に、管理ポータルでアプリケーションを素早く特定する方法について説明します。

API の運用中に、ご自分の API にアクセスしているアプリケーションに関する情報を素早く把握しなければならない場合があります。サポートを提供する、設定を変更する、またはアプリケーションが誤動作しているので無効にする必要がある、等がその理由として挙げられます。

13.1. 必要な情報の入手

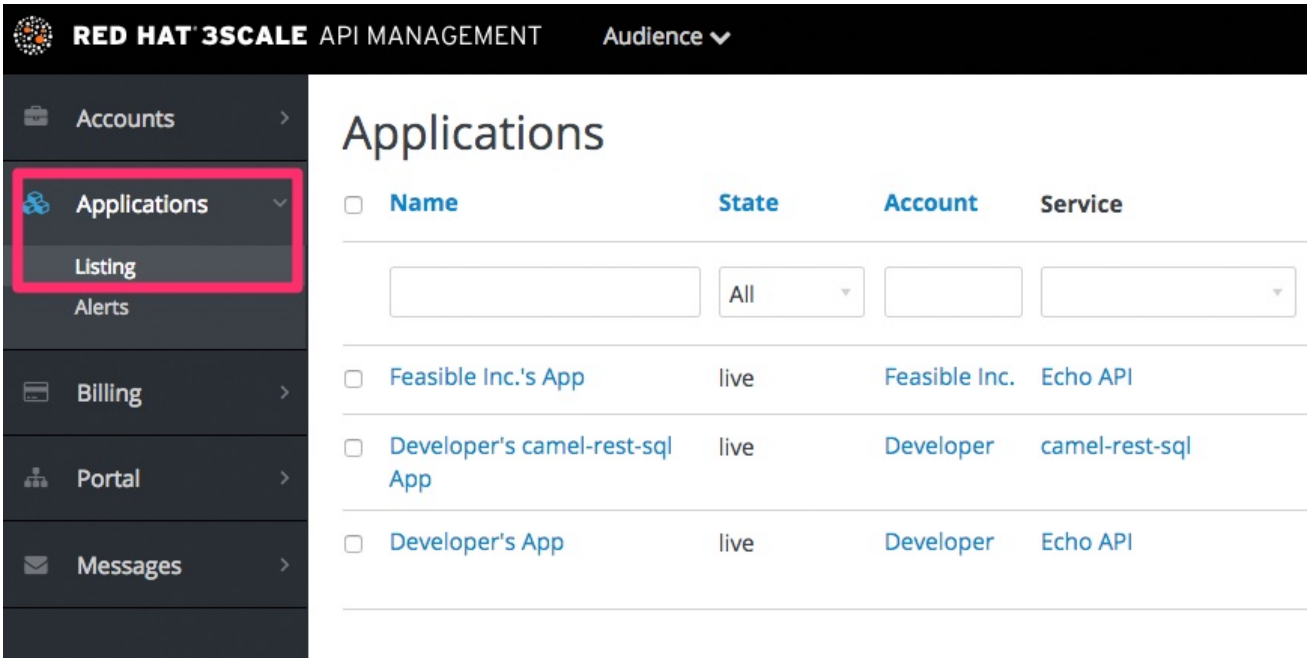
アプリケーションを特定するには、アプリケーションが属するアカウントの名前またはアプリケーションの名前が必要です。この情報がない場合には、アクセスログを確認することができます。検索を実行するには、Applications のページに移動します (**Audience > Applications > Listing**)。

認証タイプの識別子で検索する場合には、以下の情報が必要です。

- API キーのみによる認証パターン: API キー
- アプリケーション ID とアプリケーションキーによる認証パターン: アプリケーション ID (アプリケーションキーによる検索はサポートされません)
- OpenID Connect による認証パターン: client_id (シークレットによる検索はサポートされません)

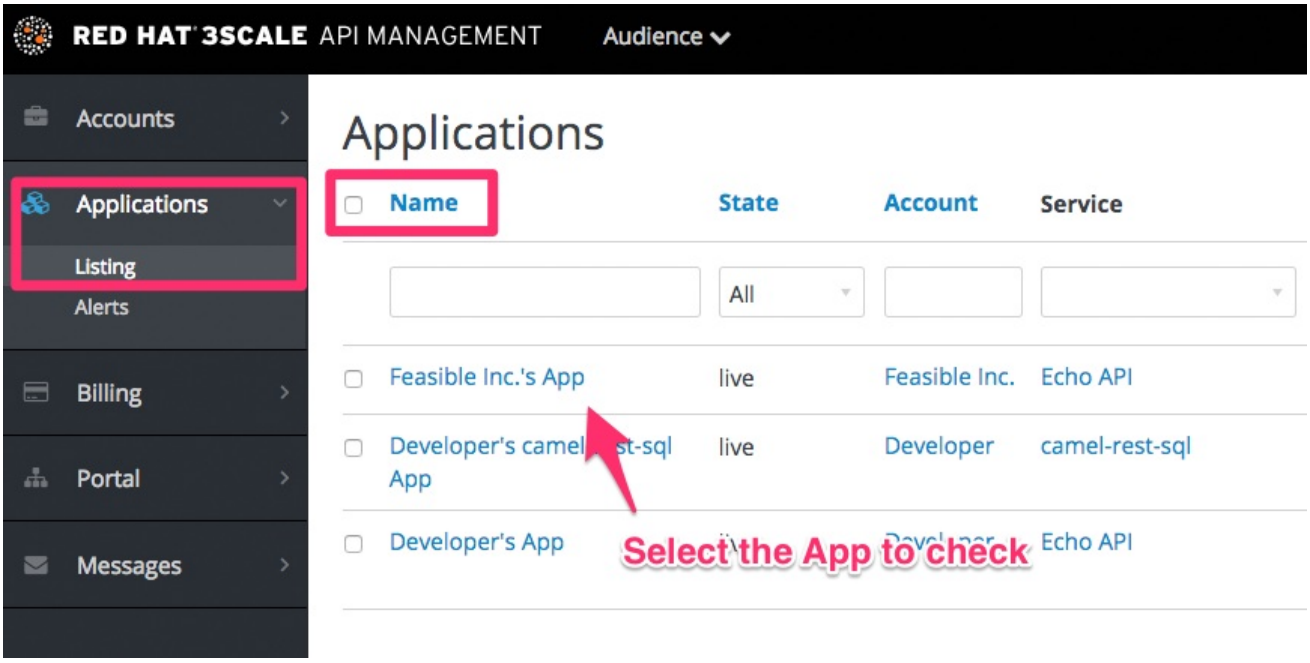
13.2. アプリケーションの検索

特定のアプリケーションを検索するには、Applications のページに移動し (**Audience > Applications > Listing**)、以下の図に示す検索ボックスを使用します。



13.3. アプリケーション情報へのアクセス

検索結果が返されたら、アクセスするアプリケーションをクリックすると、そのアプリケーションのホームページが表示されます。ホームページには、以下の図に示すような情報が含まれています。

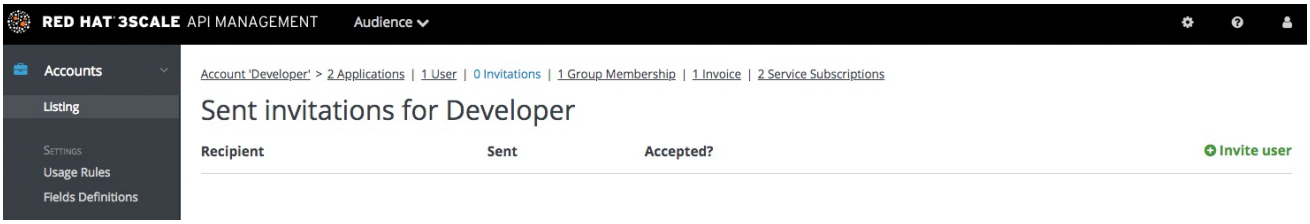

RED HAT 3SCALE API MANAGEMENT
Audience
Accounts
Applications
Listing
Alerts
Billing
Portal
Messages
Applications
Name
State
Account
Service
All
Feasible Inc.'s App
live
Feasible Inc.
Echo API
Developer's camel st-sql App
live
Developer
camel-rest-sql
Developer's App
Echo API
Select the App to check

第14章 開発者の招待

開発者アカウントに新しい開発者ユーザーを追加する方法を、以下の手順に示します。

開発者アカウントを手動で作成する場合には、管理ポータルから開発者ユーザーをそのアカウントに招待することができます。

1. **Audience** > **Accounts** > **Listing**の順に移動します。
2. 対象のアカウントを選択します。
3. Invitations を選択し、続いて **Invite user** をクリックします。



第15章 開発者のサービスからのサブスクライブ解除

管理者は、開発者のサービスへのサブスクライブを解除することができます。特定の 1 人の開発者または複数の開発者 (サービスが非推奨となった場合) に対して、この操作を実施しなければならない場合があります。

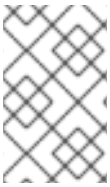
15.1. 1 人の開発者のサービスからのサブスクライブ解除

管理ポータルから、1 人の開発者のサービスへのサブスクライブを解除します。

1. 管理ポータルの Dashboard で、**Audience > Accounts > Listing > [select an account] > Service Subscriptions** の順に移動します。
2. 開発者のサブスクライブを解除するサービスの **Unsubscribe** を選択します。

15.2. 複数開発者のサービスからのサブスクライブ解除

一括処理を実施して、複数の開発者の非推奨となった/削除されたサービスへのサブスクライブを解除します。



注記

この手法を適用することができるのは、削除された/一時中断されたサービスだけです。アクティブなサービスに対して、一括のサブスクライブ解除処理を行うことはできません。

1. Dashboard で、以下に移動します。**Audience > Accounts > Subscriptions**
2. 一括でステータスの変更を行います。
3. サービスのドロップダウンメニューを使用して、開発者のサブスクライブを解除するサービスを選択します。
4. 左側のチェックボックスを使用して、サブスクライブを解除する開発者を選択します。
5. **Change state > Suspend** の順に選択し、選択した開発者のサブスクリプションを一時中断します。

Service Plans のチェックボックスを選択している必要がある点に注意してください。

第16章 アプリケーションの一時中断

本章では、アプリケーションのキーおよびアクセストークンをすべて無効にする方法について説明します。

アプリケーションがご自分の API を誤用し、他のトラフィックに影響を及ぼしている場合には、担当する開発者に連絡してコードまたは設定の修正を依頼する前に、直ちにその運用を一時中断しなければならない場合があります。

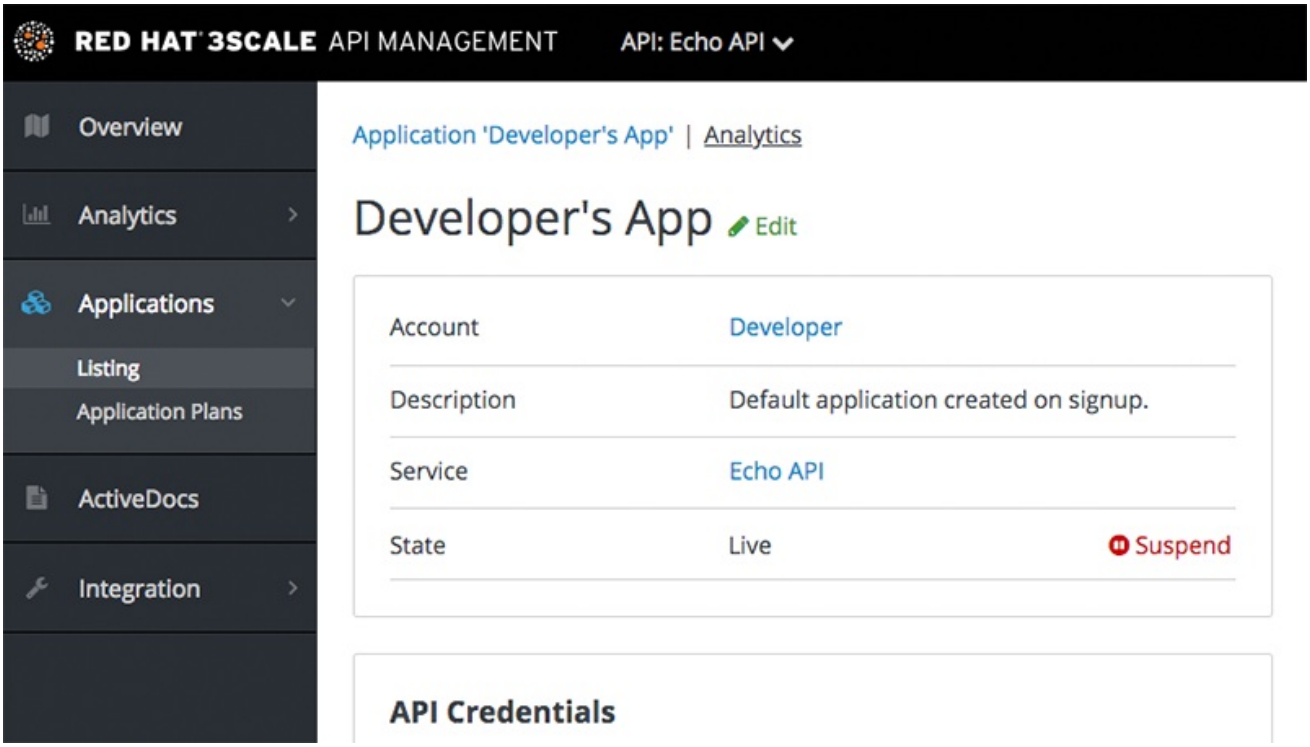
16.1. アプリケーションの特定

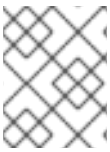
Accounts もしくは **Applications** タブから、または ここ で説明するように検索を行い、アプリケーションを特定することができます。

16.2. アプリケーションの無効化

アプリケーションを特定してアプリケーションの概要ページを表示したら、State の値の横にある Suspend アイコンをクリックします。この操作により、アプリケーションが直ちに API から無効になり、すべてのキーの動作が一時中断されます。これらのアプリケーションキーを使用した呼び出しは、制御システムによって拒否されます。

問題のある挙動が修正されたら、同じボタンを使用してアプリケーションの一時中断を解除することができます。





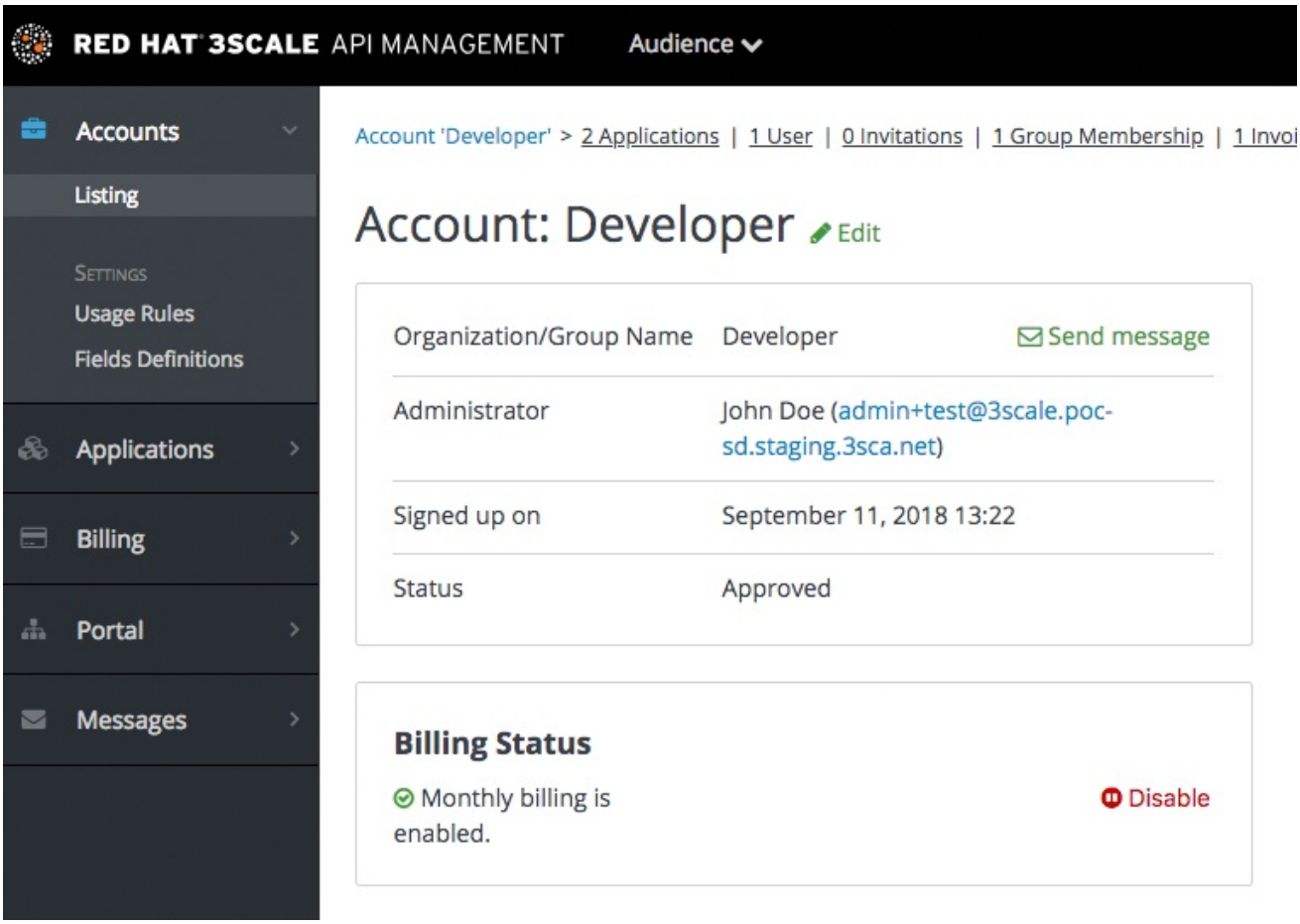
注記

エージェントでキャッシュを使用している場合には、直ちに一時中断されず若干時間を要する場合があります。

16.3. 開発者への連絡

アプリケーションの開発者への連絡方法は、ワークフローおよびポリシーによって異なります。同じ

ページでアカウント名をクリックすると、アカウントビューが表示され、アプリケーションを所有するアカウントのメイン管理者を特定することができます。電子メールにより、または以下の図に示す Send message ボタンをクリックしてユーザー用のダッシュボードメッセージを生成して、管理者に連絡を取ります。



第17章 アプリケーションの削除

管理ポータルからアプリケーションを削除するには、以下の手順に従う必要があります。

オプション 1:[Your_API_name] の全アプリケーションのリストからアプリケーションを削除する。

1. Dashboard で **[Your_API_name]** をクリックします。
2. **Overview** タブをクリックします。
3. **Overview** のページの左側パネルで、**Applications** をクリックします。
4. **Listing** を選択します。
5. アプリケーションをクリックします。
6. アプリケーションの詳細が含まれるページが表示されます。**Edit** をクリックします。
7. アプリケーションを削除するには、**Delete** をクリックします。
8. 確認メッセージが表示されます。**OK** をクリックして削除を確認します。

オプション 2:特定のアプリケーションプランに基づいてアプリケーションを削除する。

1. 管理ポータルで、**Dashboard** をクリックします。
2. **API** を選択します。
3. **Published Application Plans** セクションで、アプリケーションを選択します。
4. アプリケーションをクリックします。
5. アプリケーションの詳細が含まれるページが表示されます。**Edit** をクリックします。
6. アプリケーションを削除するには、**Delete** をクリックします。
7. 確認メッセージが表示されます。**OK** をクリックして削除を確認します。

あるいは、**Application Delete** という操作により、3scale API Docs を介してアプリケーションを削除することもできます。

第18章 API の削除

API のサービスを削除して、API を削除することができます。サービスを削除すると、API のアプリケーション、アプリケーションプラン、メトリクス、課金ルール、機能、サービスプラン、およびサブスクリプションが削除されます。

API を削除するには、以下の手順を実施します。

1. **[Your_product_name] > Overview > Edit**の順に移動します。
2. サービスの削除セクションで、リンクをクリックしてサービスを削除します。

パート III. アクセス制御

3scale API プロバイダーは、個々の使用状況の詳細を把握するメソッドを指定し、メトリクスを定義し、メソッドおよびメトリクスをマッピングルールに関連付け、価格や使用上限を設定するアプリケーションプランの作成を行い、API へのアクセスを制御します。アプリケーションプランは、メソッドおよびメトリクスに関して収集したデータを使用して使用上限を適用します。

第19章 使用状況の詳細を把握するためのメソッドの指定およびメトリクスの追加

アプリケーションプランでは、API へのコンシューマーアクセスに対して上限および課金ルールを設定します。上限および課金ルールの適用を有効にするには、個々の使用状況データを収集するメソッドを API に指定するか、メトリクスを追加します。指定した各メソッドおよびカスタムメトリクスにマッピングルールを追加します。マッピングルールは、把握する使用状況データの詳細を指定します。

プロダクトおよびバックエンドに対して、メソッドを指定したりメトリクスを追加したりすることができます。プロダクトの場合、これにより、プロダクトのアプリケーションプランに上限および課金ルールを設定することができます。バックエンドの場合、これにより、そのバックエンドをバンドルするあらゆるプロダクトのアプリケーションプランに上限および課金ルールを設定することができます。

個々の呼び出しの数を把握するメソッドを指定します。これにより、API の使用をより細かい粒度で追跡することができます。メソッドにトラフィックがレポートされると、メソッドおよび **Hits** メトリクスのカウンターが自動的に加算されます。API バックエンドのエンドポイントごとにメソッドを指定することや、エンドポイントと HTTP メソッドの組み合わせにメソッドを指定することができます。API のエンドポイントをここで追加するメソッドにマッピングする方法については、メソッドおよびメトリクスへのマッピングルールの追加 を参照してください。

メトリクスは、プロダクトとバックエンドの両方について、API の使用状況を追跡するのに適しています。**Hits** は、各 API に存在する組み込みのメトリクスです。これは、API への呼び出しの数を追跡します。**Hits** とは別に API の使用状況を把握するには、使用状況を別の単位でレポートするメトリクスを定義します。この単位は数値化が可能で、ビジネス目標に対して意味を持つものでなければなりません (例: メガバイト数、CPU 時間、または API によって返される要素数など)。CPU 時間や **mb** などの **Hits** 以外のメトリクスは、デフォルトでは提供されません。ユーザーが設定した外部サービスによって呼び出されるエンドポイントを使用して、これらのメトリクスを取得します。

メソッドおよびメトリクスは、API をパッケージ化するための骨組みです。それぞれのアプリケーションプランにより、指定した各メソッドおよびメトリクスに、異なる使用上限および課金ルールを定義することができます。メトリクスおよびメソッドにより報告される使用状況についての詳細は、API 解析 を参照してください。

19.1. プロダクトおよびバックエンドへのメソッドの追加

プロダクトまたはバックエンドにメソッドを追加すると、個々の使用状況の詳細を把握する API にメソッドを設定できます。アプリケーションプランにより、プロダクトまたはバックエンドに追加する各メソッドに上限を設定することができます。プロダクトにメソッドまたはメトリクスを追加する手順は、メソッドまたはメトリクスをバックエンドに追加する手順と類似しています。

手順

1. **[Your_product_name] > Integration > Methods & Metrics** または **[Your_backend_name] > Methods & Metrics** の順に移動します。
2. **Add a method** をクリックします。
3. **Friendly name** フィールドに、メソッドの簡単な説明を入力します。この名前は、3scale 管理ポータルのさまざまなセクションに表示されます。平易な名前は、プロダクト単位で一意でなければなりません。

重要

メソッドのシステム名を変更する場合やそれらを削除する場合は、注意してください。メソッドの変更前のシステム名/削除したメソッドをポイントするマッピングルールがあると、これらの変更によりデプロイ済みの 3scale インテグレーションが機能しなくなります。

4. **System name** フィールドに、3scale Service Management API 経由で使用状況を報告するのに使用する API のメソッドの名前を入力します。
システム名の形式を決定します。この名前はエンドポイントと同じにするか (**/status**)、たとえばメソッドおよびパスを含めることができます (**GET_/status**)。ただし、システム名は、以下のルールに準拠する必要があります。

   - プロダクトまたはバックエンド単位で一意である。
   - 英数字、アンダースコア (_)、ハイフン (-)、またはスラッシュ (/) のみが含まれる。
   - スペースなし。
   - バックエンド API の場合、メソッドのシステム名には、マップされたバックエンドを識別する数値の文字列が含まれます。このバックエンド識別子を変更することはできません。

5. オプション:**Description** フィールドに、メソッドのより詳細な説明を入力します。
6. **Create Method** をクリックします。

検証手順

- 追加したメソッドがアプリケーションプランで利用可能である。

次のステップ

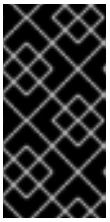
- **[Your_product_name] > Applications > Application Plans > [plan_you_want_to_edit]**の順に移動して、各メソッドの上限および課金ルールを編集する。

19.2. プロダクトおよびバックエンドへのメトリクスの追加

メトリクスを追加して、API へのすべての呼び出しについて把握する使用状況の単位を指定します。アプリケーションプランにより、プロダクトまたはバックエンドに追加する各メトリクスに上限を設定することができます。プロダクトにメソッドまたはメトリクスを追加する手順は、メソッドまたはメトリクスをバックエンドに追加する手順と類似しています。

手順

1. **[Your_product_name] > Integration > Methods & Metrics**または **[Your_backend_name] > Methods & Metrics** の順に移動します。
2. **Metrics** タブをクリックします。
3. **Add a metric** をクリックします。
4. **Friendly name** フィールドに、メトリクスの簡単な説明を入力します。この名前は、3scale 管理ポータルのさまざまなセクションに表示されます。平易な名前は、プロダクト単位で一意でなければなりません。



重要

メトリクスのシステム名を変更する場合やそれらを削除する場合は、注意してください。メトリクスの変更前のシステム名/削除したメトリクスをポイントするマッピングルールがあると、これらの変更によりデプロイ済みの 3scale インテグレーションが機能しなくなります。

5. **System name** フィールドに、3scale Service Management API 経由で使用状況を報告するのに使用する API のメトリクスの名前を入力します。
システム名の形式を決定します。ただし、システム名は、以下のルールに準拠する必要があります。

    - プロダクトまたはバックエンド単位で一意である。
    - 英数字、アンダースコア (_)、ハイフン (-)、またはスラッシュ (/) のみが含まれる。
    - スペースなし。
    - バックエンド API の場合、メトリクスのシステム名には、マップされたバックエンドを識別する数値の文字列が含まれます。このバックエンド識別子を変更することはできません。

6. **Unit** フィールドに単位を入力します。

    - **hit** のように、単数形の名詞を使用します。解析チャートでは、単数形が複数形に変わります。

7. オプション:**Description** フィールドに、メトリクスのより詳細な説明を入力します。

8. **Create Metric** をクリックします。

検証手順

- 追加したメトリクスがアプリケーションプランで利用可能である。

次のステップ

- **[Your_product_name] > Applications > Application Plans > [plan_you_want_to_edit]**の順に移動して、各メトリクスの上限および課金ルールを編集する。
- **[Your_product_name] > Integration > Mapping Rules**の順に移動して、メトリクスを 1 つまたは複数の URL パターンにマッピングする。メソッドおよびメトリクスへのマッピングルールの追加 を参照してください。

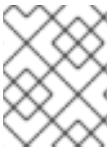
19.3. メソッドおよびメトリクスをインポートするその他の方法

API に複数のエンドポイントがある場合、メソッドを自動的に指定し、3scale プロダクトおよびバックエンドにメトリクスを追加する方法は 2 つあります。

- Swagger 仕様を使用したインポート

19.4. メソッドおよびメトリクスへのマッピングルールの追加

マッピングルールは、以前プロダクトおよびバックエンドに作成した メソッド および メトリクス にマッピングされる操作です。



注記

マッピングルールは、以前に作成したメソッドに必要ですが、メトリクスにはオプションになります。

手順

1. **[Your_product_name] > Integration > Mapping Rules**の順に移動します。
2. **Create mapping rule** をクリックします。
3. **Verb** フィールドには HTTP メソッド **GET** が事前設定されていますが、ドロップダウンリストから他のオプションを選択できます。
4. **Pattern** フィールドに、スラッシュ (/) で始まる有効な URL を追加します。URL は、中かっこ **{}** 内に指定したワイルドカードから設定できます。
5. **Metric or method to increment**フィールドで、以前に作成したメソッドまたはメトリクスのいずれかを選択します。
6. **Increment by** フィールドには **1** が事前設定されていますが、必要に応じて変更してください。
7. **Create mapping rule** ボタンをクリックします。

検証手順

- マッピングルールを確認するには、**[Your_product_name] > Integration > Methods & Metrics** の順に移動します。各メソッドおよびメトリクスの **Mapped** 列に、チェックマークが付いているはずです。

19.5. 関連情報

- API プロダクトおよび API バックエンドの詳細は、Getting started with 3scale を参照してください。

第20章 アプリケーションプラン

アプリケーションプランでは、API の利用者に許可するアクセス権限のさまざまなセットを定義します。これらのプランで、流量制御からアクセス可能なメソッドまたはリソース、有効な機能に至るまで、あらゆる設定を定義することができます。

20.1. アプリケーションプランの作成方法

デフォルトでは、3scale アカウントが作成されると、2 つのプランが設定されます。つまり、Basic および Unlimited です。これらのプランを維持して編集することも、専用のプランを作成することもできます。必要な数だけプランを作成することができます。

新たなアプリケーションプランを作成するには、以下の手順に従います。

1. **[Your_product_name] > Applications > Application Plans**の順に移動します。
2. **Create Application Plan** のボタンをクリックします。
3. **Create Application Plan** のページで、新しいプランの名前およびシステム名を入力します (システム名は一意でなければなりません)。
4. アプリケーションがご自分の API にアクセスする前に承認を要求する場合には、**Applications require approval?** のチェックボックスを選択します。

プランを作成したら、流量制御 をプロビジョニングし、有料プラン を設定することができます。

20.2. デフォルトのアプリケーションプランの設定

すべてのプランを作成したら、ユーザーがサインアップしてアプリケーションを登録する際のデフォルトプランを選択することができます。

デフォルトのアプリケーションプランを選択するには、以下の手順を実施します。

1. **[Your_product_name] > Applications > Application Plans**の順に移動します。
2. **Default Plan** セクションのドロップダウンリストからプランを選択します。

デフォルトのアプリケーションプランを指定しないと、新規ユーザーがご自分の API にアクセスするためにサインアップしても、デフォルトではアプリケーションは作成されません。つまり、ユーザーは API にアクセスすることができません。

第21章 有料プランのプロビジョニング

API (プロダクトまたはバックエンドのいずれか) を収益化する最も一般的な方法の1つは、使用量に応じたサブスクリプション料を定義することです。本セクションでは、アプリケーションプランを使用して課金レイヤーをプロビジョニングする方法、および有料プランを設定する方法について説明します。プロダクトおよびバックエンドレベルに加えて、アカウントレベルで課金ルールを適用することもできます。これらのトピックは、本章の以降の項目で説明します。

21.1. 課金モデルの決定

初めに決めなければならないことは、課金モデルでレイヤー間を区別する方法です。レイヤーは、ボリュームまたは使用量、API 機能、他のリソースへのアクセス、またはこれらの組み合わせに関連付けることができます。

- **ボリューム/使用量。**ボリュームに基づいてレイヤー間を区別するのが最も一般的な方法です。通常ボリュームは、顧客に提供する価値およびサービスのコストと強い相関があるためです。プロダクトへの呼び出しにグローバルなヒットカウントを適用することや、メソッドレベルでより細かな粒度の測定を適用することができます。**ボリュームドライバー** は、グローバルな hits メトリクスレベルで、または hits 下の個々のメソッドに適用されます。どのメトリクスにも、複数の課金ルールを適用することができます。ヒット数の計算は、1カ月の請求サイクルにわたって累積される点に注意してください。

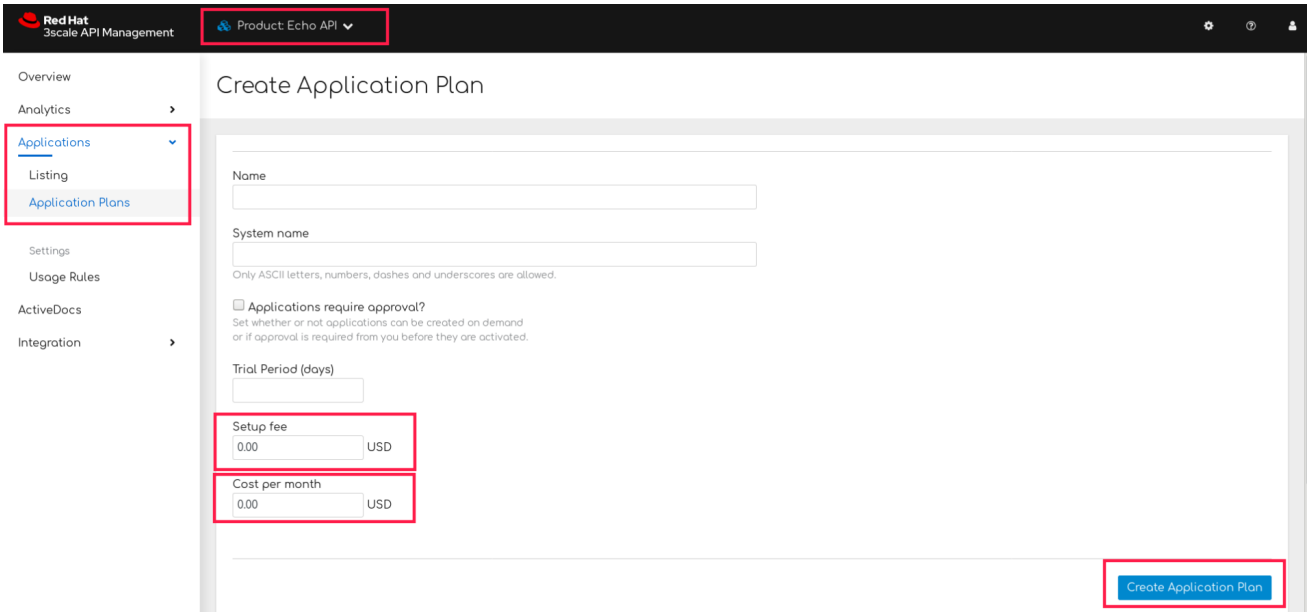
- **機能。**レイヤーに応じて、プロダクトの特定部分へのアクセスを有効または無効にすることができます。これは、標準レベルとプレミアムレベルを区別するのに適した手法です。

- **リソース。**他のリソース (顧客に価値を提供したり、ご自分のインフラストラクチャーでの費用発生の原因になったりする) へのアクセスに基づいて、レイヤーを作成することもできます。リソースの例としては、消費した帯域幅のギガビット数、ユーザーの数、またはトランザクションの値などが挙げられます。**リソースドライバー** はボリュームドライバーと類似していますが、カスタムメトリクスに適用されます。

課金ドライバーを決定したら、レイヤーが定額サブスクリプション、変動レート、または1回限りの前払い料金のいずれに基づくかを決定する必要があります。上記の 3 つの課金ドライバーは、すべて1回限りの前払い料金または1カ月の定額サブスクリプションに対応しています。ヒット数またはリソースの消費量に基づいて課金することを選択した場合には、当然課金は変動要素を持ちます。

21.2. アプリケーションプランへの課金ルールの設定

新たなアプリケーションプランを作成することも、既存のプランを編集することもできます。新規アプリケーションプランを作成する場合には、前払い料金または定額サブスクリプションを自由に設定することができます。



アプリケーションの編集ビューで、前払い料金および定額サブスクリプションを入力または変更することができます。

次に、「課金モデルの決定」で選択した課金ドライバーを設定します。

1. **[Your_product_name] > Overview > Applications > Application Plans**の順に移動します。
2. アプリケーションプランの名前をクリックします。
3. **Metrics, Methods, Limits & Pricing Rules**セクションに移動します。ここで、総ヒット数のレベルで、メソッドのレベルで、またはその他のカスタムメトリクスに対して、課金ルールを定義することができます。

いくつかの課金ドライバーがすでにメトリクスとして存在する場合には、その項目を編集することができます。

課金ルールの設定が完了したら、**Update Application plan** をクリックします。

21.3. 追加課金レイヤーの作成

1 つのアプリケーションプランを使用して、API 有料プランを定義することができます。すべての課金ルールがボリュームまたはリソースドライバーにより定義される場合は、通常これに該当します。ただし、開発者コミュニティーのグループごとに別のプランを提供する場合には、アプリケーションプランをさらに追加します。

そのための最も簡単な方法は、最初のアプリケーションプランをアプリケーションプランの概要ページからコピーすることです。これにより、既存のメトリクスおよび課金ルールが、すべて事前に入力されます。最初に完全なプランを作成するのに多くの注意を払うほど、プランのコピー機能により多くの時間を節約することができます。

21.4. 有料プランのプロビジョニング

プランをプロビジョニングするには、開発者は新しいアプリケーションを作成し、新しい有料プランの 1 つを選択する必要があります。管理コンソールから、開発者に代わってこの作業を行うこともできます。既存のアプリケーションに関して、既存のプランから新しい有料プランのいずれかに変更することもできます。

21.5. その他の参考情報

定額の課金プランに加えて、流量制御を使用してレイヤー間を区別するのも一般的です。これについては、流量制御のプロビジョニング で説明します。

第22章 流量制御のプロビジョニング

流量制御により、API リソース (プロダクトおよびバックエンド) へのアクセスにスロットリングを適用することができます。アプリケーションプランを使用して、開発者グループごとに異なる制限を設定することができます。

流量制御の設定が完了すると、これらの制限により、開発者が 3scale バックエンドに承認を要求する呼び出しを行う際に受け取るレスポンスがコントロールされます。

22.1. アプリケーションプランの設定

アプリケーションプランをまだ定義していない場合には、まず始めにプランを作成します。定義している場合には、流量制御を設定するプランを選択して、**edit** をクリックします。

アプリケーションプラン作成の詳細は、アプリケーションプラン を参照してください。

22.2. 流量制御の設定

流量制御を設定するには、以下の手順を実施します。

1. **[Your_product_name] > Overview > Applications > Application Plans**の順に移動します。
2. 設定するアプリケーションプランの名前をクリックします。
3. **Metrics, Methods, Limits and Pricing Rules**までスクロールダウンします。
4. **Limits** をクリックします。
5. プロダクトまたはバックエンドレベルで制限を設定します。
6. 必要な制限の設定が完了したら、**Update Application plan** をクリックして変更を保存します。

22.3. 新しい流量制御の適用

流量制御を定義したので、以下のような状況になります。

- アラートを設定している場合には、通知を送信する時期を決定するのに新しい制限が使用されます。
- 3scale バックエンドへの呼び出しの数を超えると、制限が考慮され、関連するエラーメッセージが表示されます。APIcast のエラーメッセージに関する詳細は エラーメッセージの設定 を参照してください。

流量制御が有効になると、ダッシュボードに制限に達しつつあるユーザーが表示され、プランをアップグレードする可能性のあるユーザーをすばやく簡単に確認することができます。ソフトな制限およびハードな制限の詳細は、**スタートガイド** で高度なパスの アプリケーションプランによる API アクセスポリシーの設定 を参照してください。

22.4. 補足情報

流量制御を設定する以外に、同じメトリクスに対して変動課金ルールを設定することもできます。有料プランのプロビジョニング を参照してください。

パート IV. 請求

第23章 請求設定の定義

本項では、Red Hat 3scale での請求機能の仕組みについて説明します。

請求設定は **Charging & Gateway** および **Credit Card Policies** からなり、共に管理ポータルの **Audience > Billing** で設定することができます。

23.1. MODE (CHARGING & GATEWAY)

3scale での請求は暦月に基づき、**前払い** および **後払い** の設定が可能です。

- 前払いモードでは、すべての固定費および開設費が月の初めまたはあん分した請求期間の初めに請求されます。変動費は、必ず翌月の初めに計算され請求されます。
- 後払いモードでは、すべての固定費および変動費が翌月の初めに請求されます。

詳細は、自動請求プロセス を参照してください。

23.2. CHARGING ENABLED (CHARGING & GATEWAY)

この設定により、クレジットカードを使用した支払いが有効になります。このチェックボックスが選択されていると、期限を迎えたすべての請求書に対する課金が、選択した支払いゲートウェイを使用して行われます。このチェックボックスを未選択のままにすると、請求が行われ請求書は発行されますが、実際の支払い行為は発生しません。

23.3. CURRENCY (CHARGING & GATEWAY)

請求およびクレジットカードによる支払いを行う際の通貨を選択します。選択した通貨が支払いゲートウェイでサポートされていることを確認してください。これはグローバルな設定で、すべての請求書および支払いに適用されます。開発者アカウントごとに異なる通貨を設定することはできません。

23.4. INVOICE FOOTNOTE (CHARGING & GATEWAY)

Invoice footnote フィールドに入力されたテキストは、すべての PDF ファイルの請求書の最下部に表示されます。このフィールドを使用して、顧客の課金および請求ポリシーに関する補足情報を提供することができます。

脚注のテキストが変更されても、PDF ファイルがすでに生成されている請求書には自動的に適用されません。ただし、請求書の PDF ファイルを再生成することにより、変更を適用することが可能です。

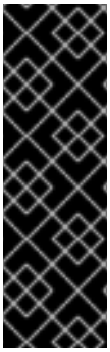
23.5. TEXT TO SHOW IF VAT/SALES TAX IS 0% (CHARGING & GATEWAY)

請求を行うアカウントの VAT/消費税が 0% の場合に、このフィールドを使用して PDF ファイルの請求書にテキストメッセージを追加します。この行は、VAT/消費税が明示的に 0 に設定されている場合にのみ表示され、それ以外の場合には表示されません。このテキストは、管理ポータルの Invoice のページにも表示されます。

この設定の詳細は、VAT/消費税 セクションを参照してください。

23.6. 通貨用 YAML の設定

currencies.yml ファイルにより、3scale デプロイメントの通貨リストを設定することができます。3scale では、ISO 4217 に基づく 3 文字の通貨コードが使用されます。



重要

- 選択した通貨が支払いゲートウェイでサポートされるようにしてください。
- 3scale では、クレジットカードによる支払いに関して以下の支払いゲートウェイとのインテグレーションが提供されます。
  - Braintree
  - Stripe

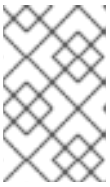
23.6.1. OpenShift での通貨設定の変更

通貨設定を変更するには、以下の手順を実施します。

手順

1. **system** config map のエントリーとして、**currencies.yml** の新しいコンテンツのソースを追加します。デフォルトの通貨リストに新たな通貨 **ARS: アルゼンチンペソ** を追加する方法を、以下の例で説明します。

```
oc patch configmap system --type merge -p "{\"data\": {\"currencies.yml\": \"production:\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\"}}"
```



注記

currencies.yml 設定ファイルのコンテンツの例を確認するには、デフォルトの YAML ファイル **currencies.yml** にアクセスします。このファイルは、新規 3scale デプロイメントのデフォルト設定を示します。

```
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. **system-(app|sidekiq)** DeploymentConfig の **system-config** ボリュームに、新規 **ConfigMap** エントリー **currencies.yml** を追加します。これにより、新しいコンテンツが該当するコンテナー内にマウントされ、新しい設定がアクティベートされます。

```
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. 新しい通貨の確認

3scale 管理ポータルで通貨が利用できることを確認するには、以下の手順を実施します。

手順

1. **Audience > Billing > Charging & Gateway**の順に移動します。
2. **Currency** ドロップダウンリストから通貨のリストが利用できることを確認します。
3. 使用する通貨を選択します。

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

3scale の請求書には、2 種類の ID があります。

実際の ID

データベース内で請求書を一意に識別する。数値による ID で、請求書の URL に使用される (例: https://<dashboard_domain>/buyers/accounts/<acount_id>/invoices/<invoice_id>) と共に、Billing API で請求書 ID として使用されます。

平易な ID

請求書に表示される、人間にとって分かりやすい ID。平易な ID が重複していても技術的には問題ありませんが、3scale アカウント内で一意にすべきです。3scale が ID の重複を認識している場合は、以下のような警告メッセージが表示されます。**この請求書 ID はすでに使用中のため、おそらく変更する必要があります**。平易な ID が 24 時間以上 **fix** と表示される場合には、サポートにご連絡ください。

以下の設定により、請求書の平易な ID のフォーマットを定義します。

monthly (デフォルト)

YYYY-MM-XXXXXXXX: ID には、年、月、および請求書の番号が含まれます。以下に例を示します。2018-02-00000013

yearly

YYYY-XXXXXXXX: 年および番号だけが含まれます。以下に例を示します。2018-00000001

Billing periods for invoice identifiers を **monthly** から **yearly** に変更しても、平易な ID のフォーマット

が変わるだけです。この変更により、請求サイクルの期間が変わることはありません。3scale API Management でサポートされるのは、月額の請求だけです。経理部門からそのような指示があった場合には、請求書 ID のフォーマットを yearly に変更します。

顧客に対して年額の課金を行う必要がある場合には、請求を手作業で処理する必要があります。新たな請求書を作成し、年額の費用項目を追加することができます。年額の課金を希望する場合には、毎月請求書の生成や課金が自動的に行われないように、アプリケーションプランを無料に設定する必要もあります。

23.8. CREDIT CARD POLICIES

ここでは、以下のページへのパスを設定することができます。

- 契約条項のページ
- プライバシーポリシーのページ
- 払い戻しポリシーのページ

第24章 支払い用のクレジットカードゲートウェイ

3scale API プロバイダーは、API に対するサブスクリプションを収益化するために、クレジットカード用の支払いゲートウェイを定義します。

24.1. 3SCALE でサポートされるクレジットカードゲートウェイ

3scale では、クレジットカードによる支払いに関して以下の支払いゲートウェイとのインテグレーションが提供されます。

- Braintree
- Stripe

クレジットカードデータを収集する場合、オンライン支払いサービスは委任契約を表示する必要があります。委任契約は、デビットの支払いのために顧客がお客様に提示した許可の記録です。委任契約には、今後のサービスに対する支払いに、指定した支払い方法が使用されることを明確に記載する必要があります。Stripe および Braintree の委任契約の詳細は、関連情報に記載のそれぞれの支払いゲートウェイの外部ドキュメントリンクを参照してください。

重要

Adyen、Authorize.Net、および Ogone インテグレーションは非推奨です。これらのゲートウェイとの新たなインテグレーションはサポートされません。既存のインテグレーションはサポートされますが、完全にサポートされる支払いゲートウェイのいずれかに移行することを推奨します。Stripe および Braintree。

関連情報

- Stripe:Accept a SEPA Direct Debit payment
- Braintree:Recurring Transactions

24.2. クレジットカードゲートウェイとしての STRIPE の設定

3scale API プロバイダーとして、Stripe を支払いゲートウェイとして管理ポータルおよびデベロッパーポータルを設定し、クレジットカードゲートウェイとして Stripe を使用して API に対するサブスクリプションからの支払いを受け取ります。

前提条件

- Stripe アカウントが必要である。
  - Stripe では、ビジネスまたはプロジェクトごとに個別の Stripe サブアカウントを使用することをお勧めする。
  - 複数のアカウント については、Stripe のドキュメントを参照すること。
- Stripe 管理者パーミッションが必要である。

手順

Stripe を支払いゲートウェイとして 3scale を設定するには、以下の手順に従います。

1. 3scale 管理ポータルでの Billing API スコープを設定したアクセストークンの生成l

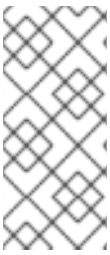
2. Stripe からのキーおよび Webhook シークレットの取得
3. 3scale 管理ポータルでの課金の設定
4. 3scale デベロッパーポータルでのクレジットカード情報の編集
5. **unsuccessfully charged** 電子メールレスポンスのテキストを更新する

24.2.1. 3scale 管理ポータルでの Billing API スコープを設定したアクセストークンの生成

1. 3scale 管理ポータルで、**Account Settings** > **Personal** > **Tokens**の順に移動します。
2. Billing API スコープを設定して **読み取り/書き込み** のトークンを作成します。
   a. **Add Access Token** をクリックします。
   b. トークンの名前を指定します。
   c. スコープを選択します。**Billing API**。
   d. パーミッションレベルを選択します。**読み取り/書き込み**。
   e. **Create Access token** をクリックします。
   f. アクセストークンをコピーします。
      - アクセストークンをファイルテキストにコピーしてください。これ以降アクセストークンは表示されません。
   g. トークンの生成を完了するには、**I have copied the token** をクリックします。

手順に戻る。

24.2.2. Stripe からのキーおよび Webhook シークレットの取得



注記

- Stripe で Webhook を設定する必要があります。
- Webhook を使用して、支払いが成功したことを 3scale に通知します。
- その後、3scale は請求書の状態を更新し、それ以上の請求を防止します。

Stripe アカウントで、**Secret Key** および **Publishable Key** を取得します。

1. Stripe ダッシュボードを開きます。
2. Stripe ドキュメントの手順に従って、API キーを確認してください。
3. **Secret Key** および **Publishable Key** をコピーします。

引き続き、Stripe アカウントで **Webhook Signing Secret** を作成します。

1. **Developers** > **Webhooks** の順に移動します。

2. **Add endpoint** をクリックします。
3. 以下のエンドポイント URL を入力します。

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

4. **Events to send** に **payment_intent.succeeded** を追加します。
5. **Add endpoint** をクリックします。
6. クリックして作成した Webhook の署名シークレットを表示し、このシークレットを書き留めます。これは **Webhook Signing Secret** です。

手順に戻る。

24.2.3. 3scale 管理ポータルでの課金の設定

3scale 管理ポータルで以下を行います。

1. **Audience > Billing > Charging & Gateway**の順に移動します。
2. **Charging enabled** を選択し、**Save** をクリックします。
3. **Credit card gateway > Gateway** で、**Stripe** をゲートウェイとして選択します。
4. 「Stripe からのキーおよび Webhook シークレットの取得」 で Stripe アカウントから取得した **Secret Key**、**Publishable Key**、および **Webhook Signing Secret** を追加します。
5. **Save** をクリックします。

手順に戻る。

24.2.4. 3scale デベロッパーポータルでのクレジットカード情報の編集

1. 開発者アカウントを使用して 3scale デベロッパーポータルにログインします。
2. **Settings > Credit Card Details** の順に移動します。
3. クレジットカード情報 (クレジットカード番号、有効期限、および CVC) を追加します。
4. **Save details** をクリックします。

手順に戻る。

24.2.5. unsuccessfully charged 電子メール応答のテキストを更新する

SCA 支払いの修正に関連して、**invoice_messenger_unsuccessfully_charged_for_buyer.text.liquid** メールのテキストには、3scale 2.10 での手動更新が必要です。

1. 3scale 管理ポータルで **Audience > Messages > Email Templates**の順に移動します。
2. **Invoice charge failure for buyer with retry**を選択します。
3. **Override** をクリックします。

4. テンプレートメッセージを更新します。これは、、課金に失敗した場合のメールレスポンスで使用される完全なテキストです。

> 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. **Create Email Template**をクリックします。

以下の手順により、**unsuccessfully charged** メールのレスポンスのメールテンプレートを更新しました。

手順に戻る。

関連情報

- Stripe Docs:API キー

24.3. BRAINTREE のクレジットカードゲートウェイとしての設定

3scale API プロバイダーは、Braintree を支払いゲートウェイとして管理ポータルおよびデベロッパーポータルを設定し、Braintree を使用して API に対するサブスクリプションからの支払いを受け取ります。

前提条件

- Braintree のアカウントが必要である。
- 3D Secure(3DS) を使用してお客様の安全なチェックアウトを確保する場合は、3scale 用に 3DS を有効にする前に、Braintree アカウントで 3DS を有効にする必要がある。
  - デフォルトでは、3scale と Braintree の両方で 3DS が **off** (無効) になっている。

手順

Braintree を支払いゲートウェイとして 3scale を設定するには、以下の手順に従います。

1. Braintree からのキーおよび業者 ID の取得
2. 3scale 管理ポータルでの課金の設定

3. 3scale デベロッパーポータルでのクレジットカード情報の編集

Braintree からのキーおよび業者 ID の取得

Braintree アカウントから **公開鍵**、**業者 ID**、および **秘密鍵** を取得します。これらの値を取得する方法の詳細は、関連情報 に一覧表示されている Braintree のドキュメントを参照してください。

手順に戻る。

3scale 管理ポータルでの課金の設定

3scale 管理ポータルで以下を行います。

1. **Audience** > **Billing** > **Charging & Gateway**の順に移動します。
2. **Charging enabled** を選択します。
3. 通貨を選択します。
    - 3scale Billing ページで指定した通貨タイプは、Braintree の業者アカウントで使用される通貨のタイプと一致する必要があります。
4. **Save** をクリックします。
5. **Credit card gateway** > **Gateway** の下で、**Braintree** をゲートウェイとして選択します。
6. Braintree からのキーおよび業者 ID の取得 の Braintree アカウントから取得した **公開鍵**、**業者 ID**、および **秘密鍵** を追加します。
7. 3DS を有効にするには、**3D Secure Enabled** を選択します。
8. **Save Changes** をクリックします。

手順に戻る。

3scale デベロッパーポータルでのクレジットカード情報の編集

3scale API コンシューマーは、3scale デベロッパーポータルでクレジットカード情報を追加または編集します。クレジットカードを発行したエンティティーと金額の詳細情報を照合するには、このウィンドウに一覧表示されるすべてのフィールドが必須です。

1. 開発者アカウントを使用して 3scale デベロッパーポータルにログインします。
2. **Settings** > **Credit Card Details** の順に移動します。
3. **Add Credit Card Details and Billing Address**のリンクをクリックします。
4. 支払い情報 (氏名と電話番号) を追加します。
5. クレジットカード情報 (クレジットカード番号、有効期限、および CVC) を追加します。
6. 請求先住所の詳細 (企業名、住所、郵便番号、市区町村名、県名) を追加します。次に、国を選択します。
7. **Save details** をクリックします。
8. プロンプトが表示されたら、購入するために 2 要素認証 (2FA) を実行します。たとえば、銀行が SMS 2FA オプションを有効にしている場合は、この方法を使用して認証プロセスを完了する必要があります。

手順に戻る。

関連情報

- Braintree アカウント からの **Public Key**、**Merchant ID**、**Private Key** の取得に関する情報は、Braintree Articles:Important Gateway Credentials を参照してください。

24.4. デベロッパーポータルを介した拒否された請求書の支払いの許可

3scale API プロバイダーは、デベロッパーポータルを介して拒否された請求書の支払いを許可します。これらの支払いを有効にするには、管理ポータルで **請求書** テンプレートを更新します。以下の手順は、デベロッパーポータルの既存のインスタンス用であることに留意してください。

前提条件

- 3scale の管理者権限が必要です。
- Stripe または Braintree のアカウントが必要です。

手順

デベロッパーポータルを介して拒否された請求書の支払いを許可するには、以下の手順に従います。

1. 3scale 管理ポータルで **Audience > Developer Portal > Content**の順に移動します。
2. **Root > Invoices > Show template**の順に移動して編集を行います。
3. コードの以下の行を編集します。

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

置き換えるスニペットを以下に示します。

```
<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. クレジットカードゲートウェイの問題のトラブルシューティング

Stripe または Braintree のどちらかを支払いゲートウェイとして使用する 3scale API プロバイダーは、これらのクレジットカードゲートウェイに関するいくつかの問題のトラブルシューティングを行うことができます。

Stripe

- Stripe からのデータを 3scale のデータとマッピングするには、Stripe フィールド **metadata.3scale_account_reference** を使用することができます (設定は **3scale-[PROVIDER_ID]-[DEVELOPER_ACCOUNT_ID]**)。

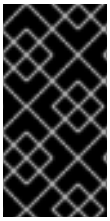
Braintree

- Braintree アカウントがサンドボックスモードにあり、何らかの問題が発生した場合には、実稼働モードに変更する必要があります。
- 3D Secure(3D) が有効になっていない 3scale デベロッパーポータルに保管されたクレジットカードの場合、3scale を Braintree と統合するための推奨されるソリューションは以下のとおりです。
  a. 3scale API プロバイダー: 3scale 管理ポータルでの課金の設定 に記載の手順に従ってください。
  b. 3scale API コンシューマー: 3scale デベロッパーポータルでのクレジットカード情報の編集 に記載の手順に従ってください。
- Braintree からのデータを 3scale のデータとマッピングするには、Braintree フィールド **customer.id** を使用することができます (設定は **3scale-[PROVIDER_ID]-[DEVELOPER_ACCOUNT_ID]**)。

24.5.1. 非推奨の支払いゲートウェイ

本セクションでは、非推奨となった支払いゲートウェイに関する一般的な情報を提供します。Adyen、OGone、および Authorize.netここで説明する情報は、既存のインテグレーションだけが対象です。これらのゲートウェイとの新たなインテグレーションはサポートされません。

24.5.1.1. Adyen インテグレーション



重要

Adyen インテグレーションは非推奨となりました。新たなインテグレーションはサポートされません。2019 年 8 月 22 日より前に存在していたインテグレーションについては、Red Hat ではサポートを提供しますが、完全にサポートされる支払いゲートウェイのいずれかに移行することを推奨します。Stripe および Braintree。

以下の手順により、アカウントの支払いゲートウェイとして Adyen を設定する方法を説明します。これにより、開発者は自分のクレジットカード情報を入力し、お客様は API へのアクセスに対する課金を計算された請求書に従って Adyen を通じて自動的に行うことができます。

有料 API の使用に対してクレジットカードへの課金を有効にする場合、支払いゲートウェイを設定することが主要なステップです。3scale アカウントで使用することのできる支払いゲートウェイは多数あります。ここでは、Adyen に関する手順を説明します。

24.5.1.1.1. 前提条件

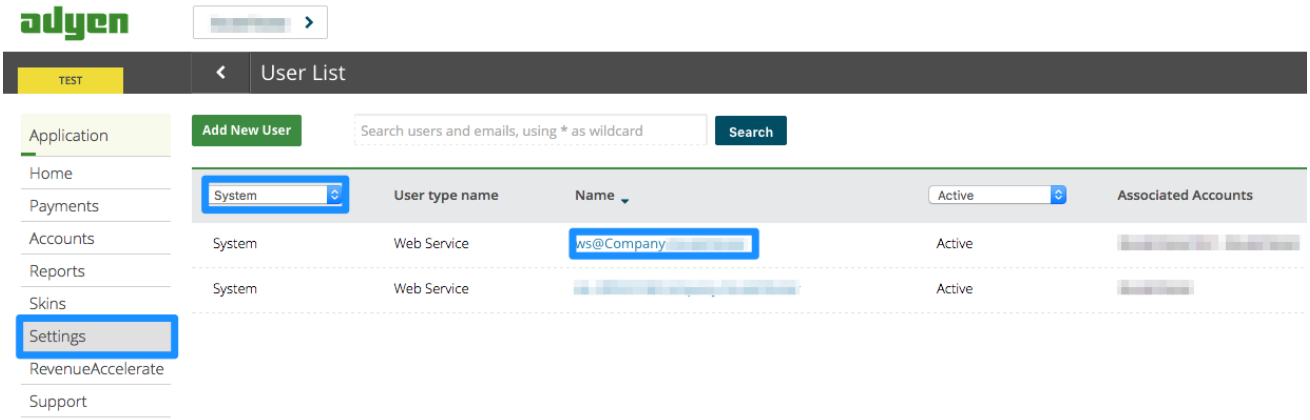
- 以下の手順を始める前に、Adyen のアカウントを作成する必要があります。
- そのサブアカウントとして、**Company** アカウントおよび **Merchant** アカウントが必要です。

- Adyen のライブアカウントを申請するには、満たさなければならない要求事項が多数あります。要求事項の詳細については、こちら をご確認ください。
- アカウントは実稼働モードでなければなりません。

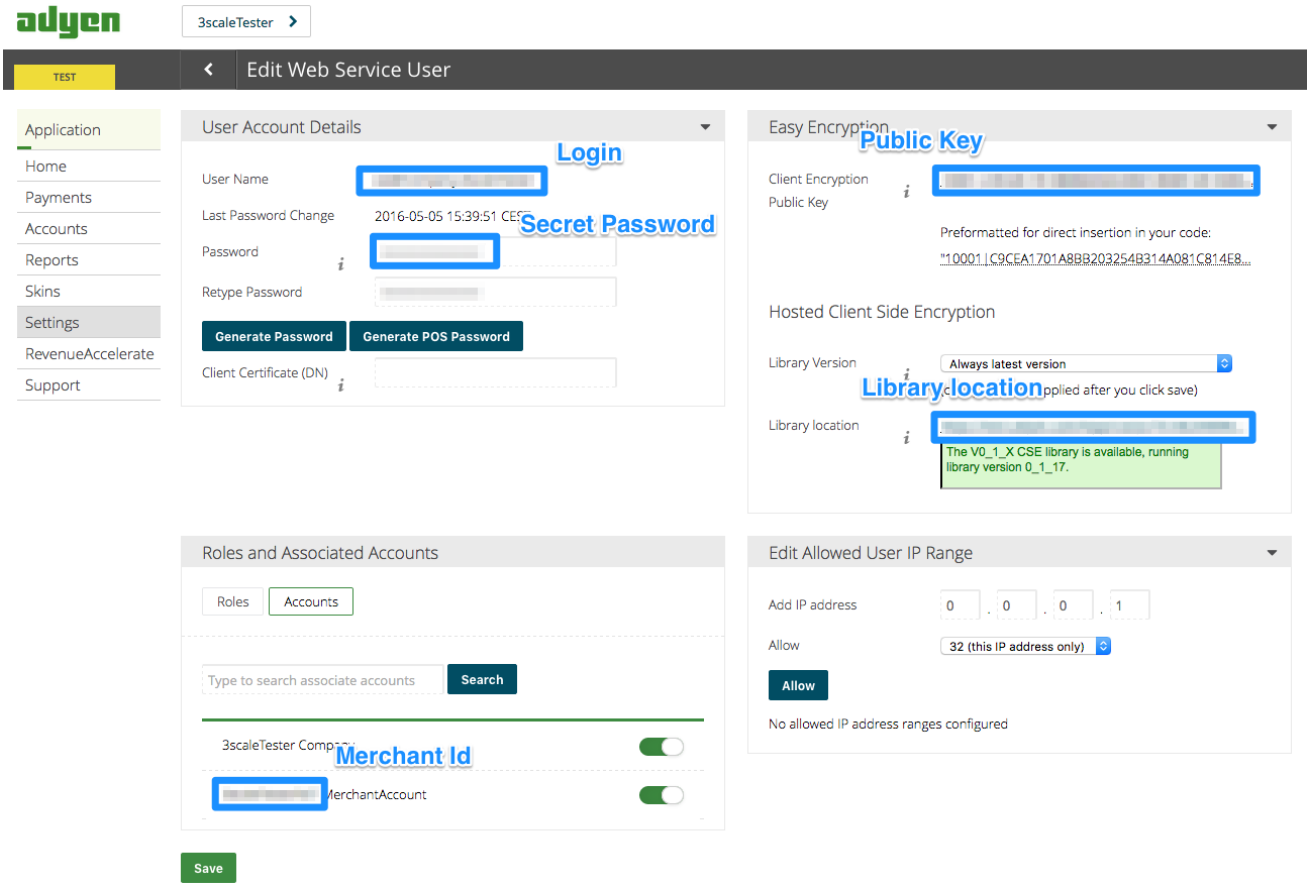
24.5.1.1.2. Adyen インテグレーションの設定

Adyen アカウント情報の確認

1. まず初めに、Adyen アカウントにログインします。次に、**Settings** > **Users** セクションでご自分のクレデンシャルを探し、以下に示すビューのドロップダウンメニューから **System** を選択します。

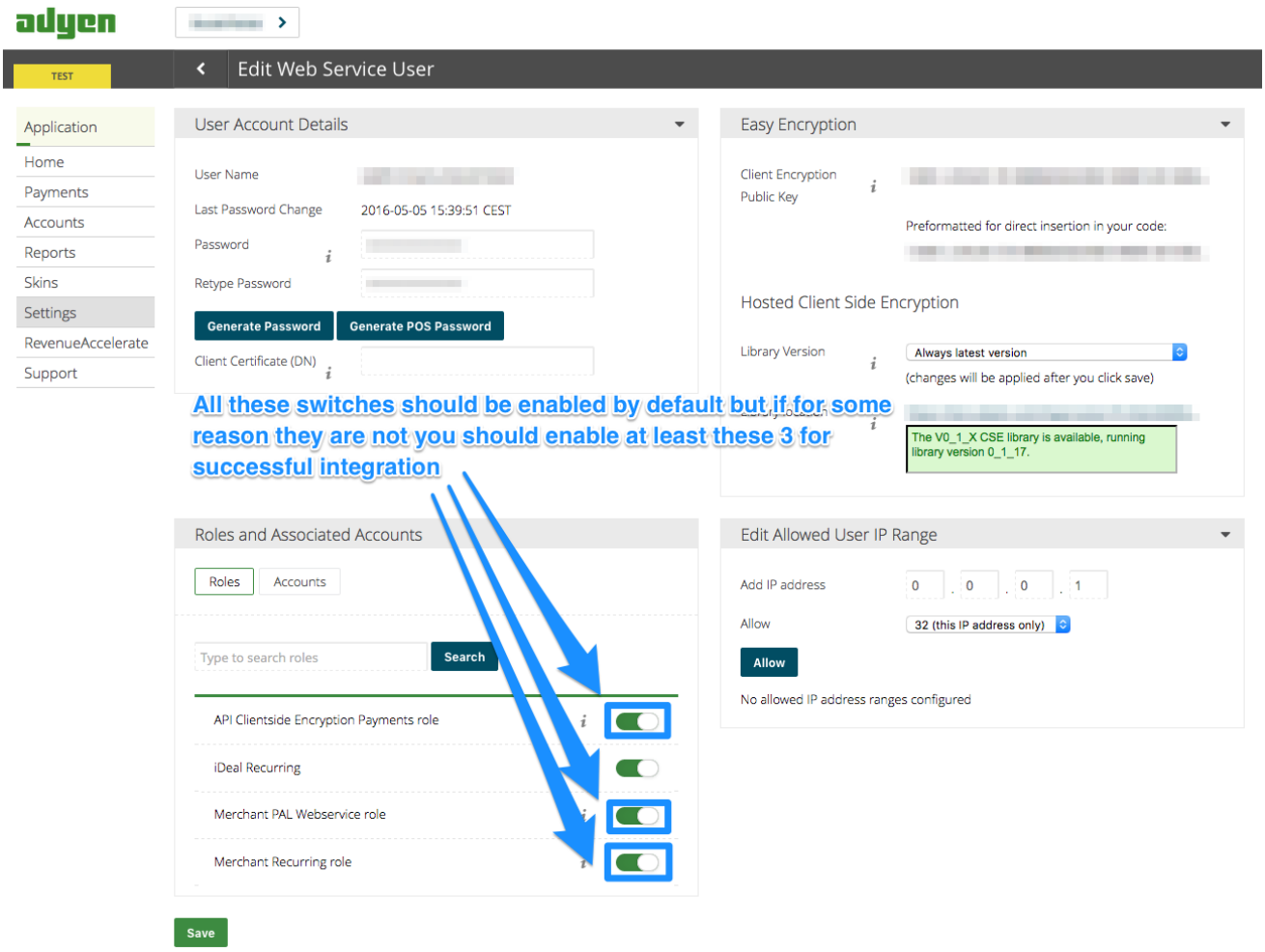


2. **Company** アカウント (リストの一番上にあるアカウント) をクリックします。これにより、Company アカウントの設定ビューに移動します。



3. こうして、**ログイン名、シークレットパスワード、クライアント暗号化用公開鍵、業者 ID**、および **ライブラリーの場所** を把握することができます。これらの情報が 3scale の請求設定で必要です。

4. **公開鍵** を表示するには、**Generate Password** をクリックしてこのパスワードをどこかにコピーする必要があります。

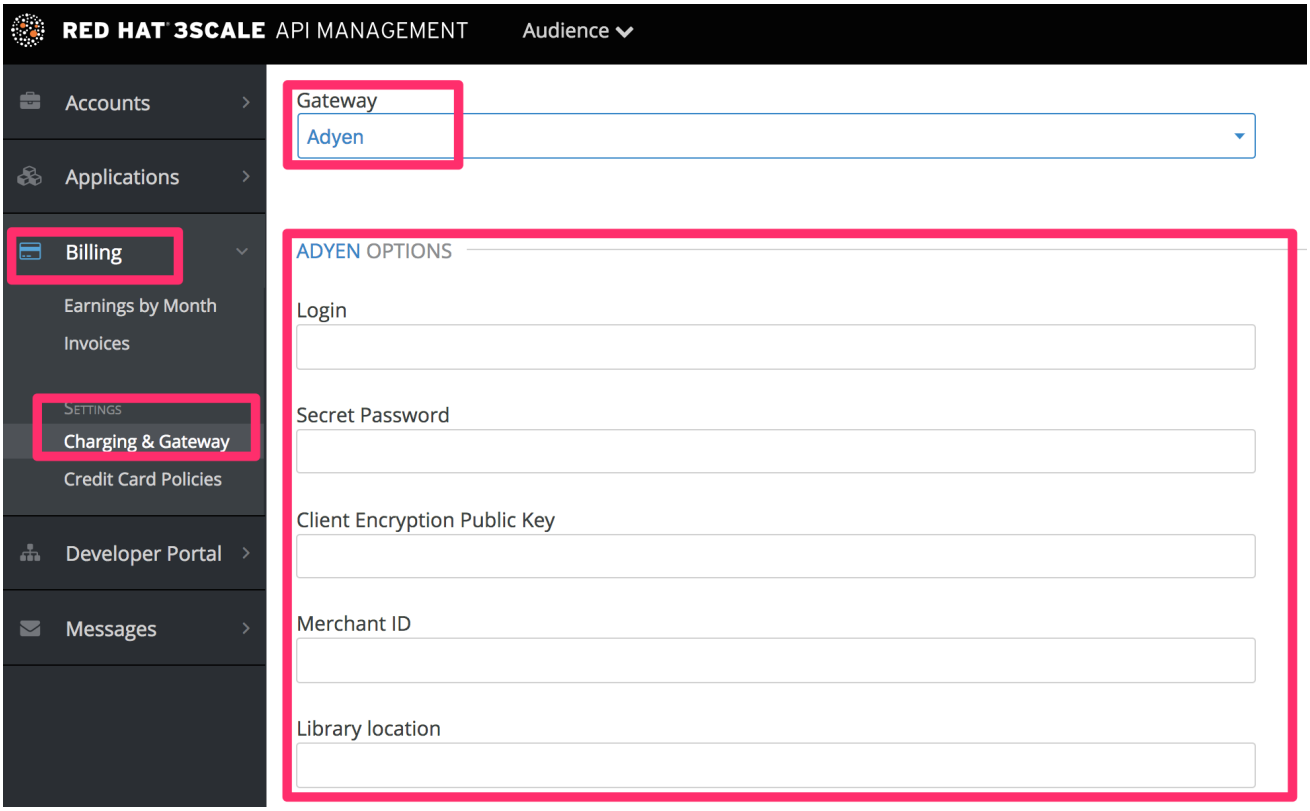


3scale アカウントでの支払いゲートウェイの設定

1. **Audience > Billing > Charging & Gateway**の順に移動し、チェックボックスを選択してクレジットカードへの課金を有効にし、**Save** をクリックします。

2. Adyen ゲートウェイへのリンクを作成するために設定しなければならないフィールドが、すべて表示されます。

3. **Gateway** のドロップダウンメニューから Adyen を選択し、変更を保存する必要があります。



Adyen API レスポンスでの alias 追加データの有効化

デフォルトでは、クレジットカード承認リクエストが 3scale から Adyen に送信されると、返されるレスポンスにはクレジットカードの一意の識別子は含まれません。

正しいクレジットカードの参照が 3scale に保存され、正しいカードに課金が行われるようにするには、この追加データを有効にする必要があります。

Adyen のサポートに連絡し、承認リクエストのレスポンスで **alias** 追加データを有効にします。

請求ワークフローのテスト

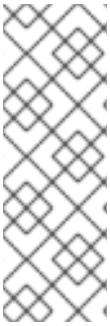
1. **前払いモード** を有効にして 1 日程度で請求書が生成されるようにし、テストサイクルを短縮します。
2. 続いて、既存のテスト用アカウントを選択し、請求用の費用項目を追加して請求書を作成します。
    a. 直ちにアカウントに対して課金を行います。
    b. このテスト法には若干のコストが発生しますが、その価値はあります。ご自分の API を使用する開発者に実際に請求を行う前に、すべてが正常に動作することを確認できるので、安心材料となります。

これで支払いゲートウェイは設定されましたが、CMS では設定されていないため、ユーザーはまだ使用できない可能性があります。Developer Portal タブに移動し、左側のナビゲーションペインでテンプレート **Payment Gateway / Show**をクリックします。

{% when "stripe" %} で始まるコードブロックの後に以下のスニペットがまだ表示されていない場合には、スニペット追加します。

```
{% when "adyen12" %}
{% if current_account.has_billing_address? %}
```

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



注記

- 2016 年 5 月 11 日より前に作成されたアカウントの場合は、上記のスニペットを手動で追加する必要があります。上記の日付以降は、デフォルトでテンプレートに含まれています。
- Adyen からのデータを 3scale のデータとマッピングするには、Adyen フィールド **shopperReference** を使用することができます (設定は **3scale-[PROVIDER_ID]-[DEVELOPER_ACCOUNT_ID]**)。

24.5.1.2. Ogone インテグレーション

重要

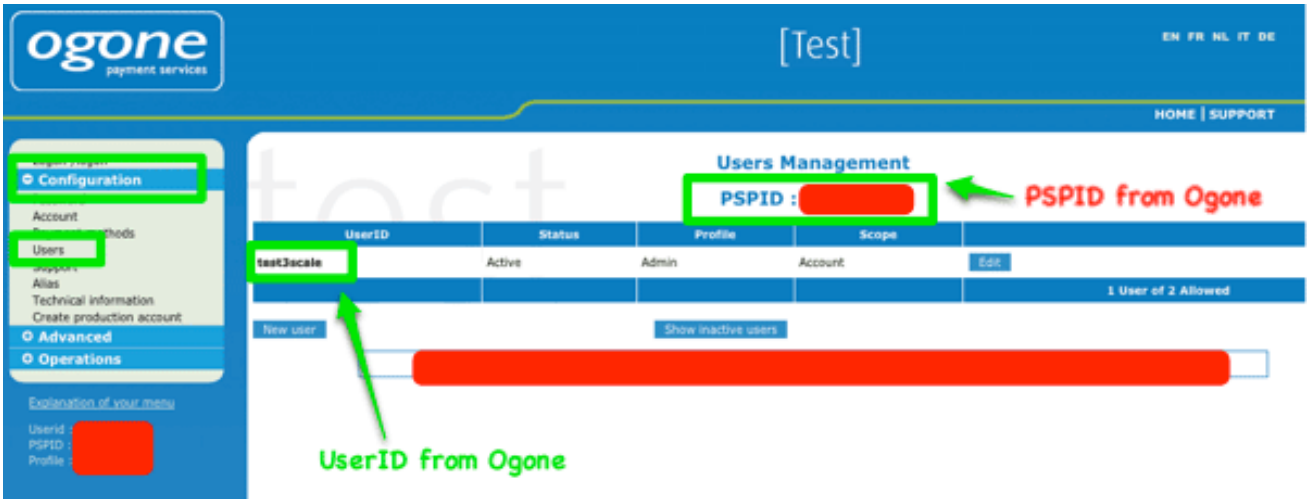
Ogone インテグレーションは非推奨となりました。新たなインテグレーションはサポートされません。2018 年 7 月 27 日より前に存在していたインテグレーションについては、Red Hat ではサポートを提供しますが、完全にサポートされる支払いゲートウェイのいずれかに移行することを推奨します。Stripe および Braintree。

API の使用に対する課金を行うために Ogone ゲートウェイを設定する手順を以下に示します。

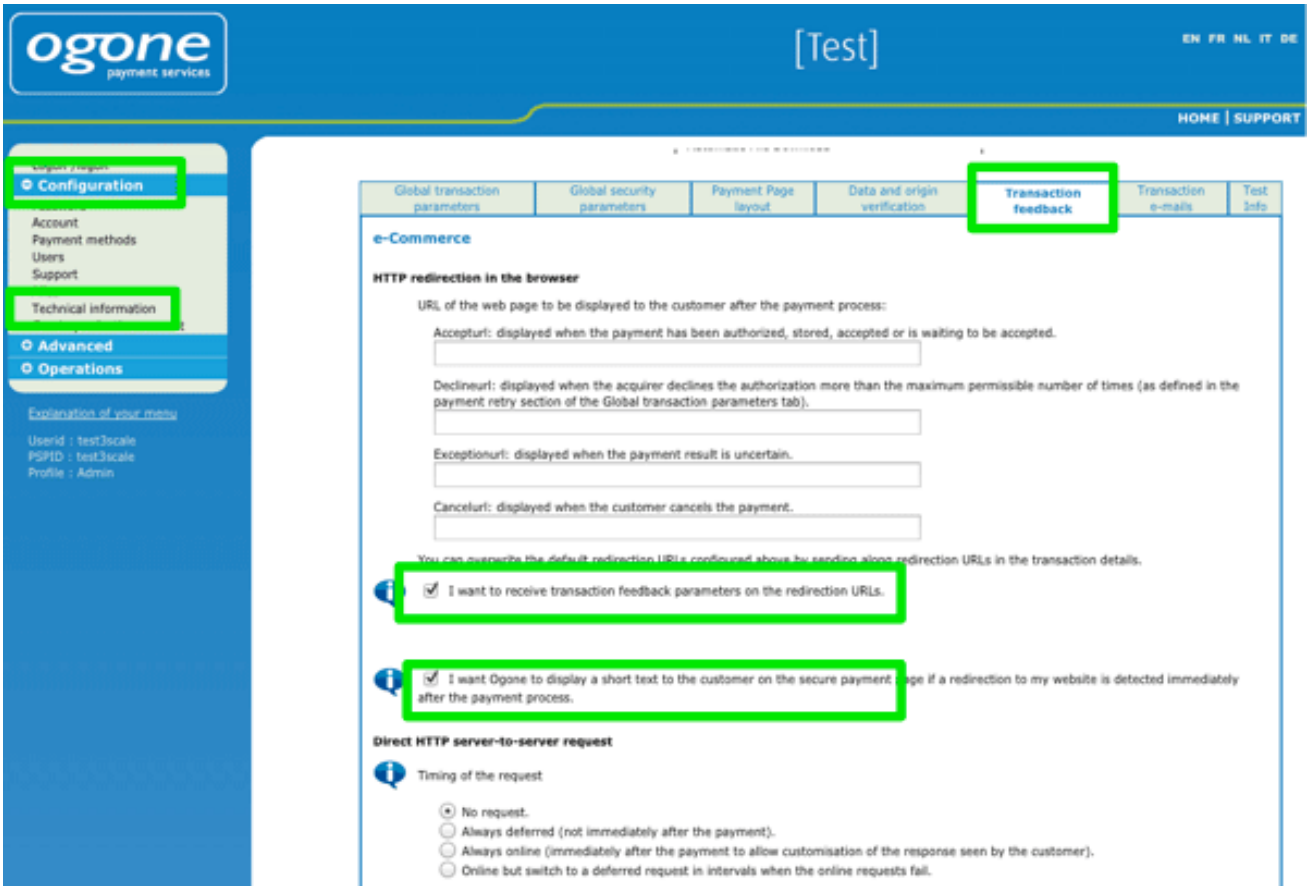
24.5.1.2.1. ステップ 1:Ogone からの API キーの取得

Ogone のアカウントを作成する必要があります。プレミアム Ogone イーコマースのアカウントが必要です (Alias Manager でアクティベートされた Horizon オプション)。これは有料オプションで、Ogone または Merchant (**backoffice > options > your options**) でアクティベートできます。

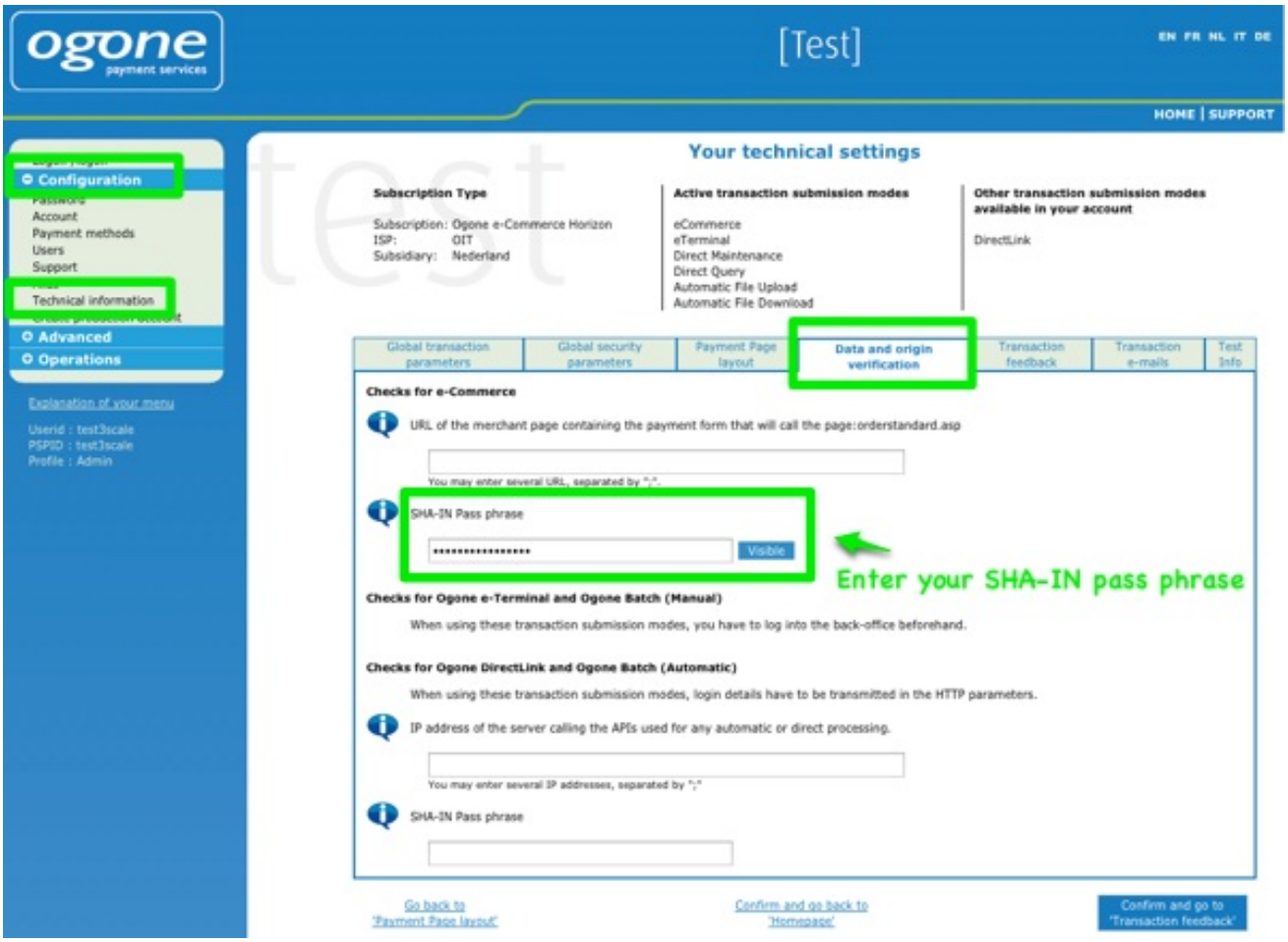
PSPID を使用して Ogone アカウントにログインします。次に、UserID は **Configuration > Users** にあります。



Ogone アカウントの技術設定がアクティブであることを確認します。Transaction Feedback ページで、SHA-OUT パスフレーズを見つけることができます。このページでは、以下に示すように 2 つの設定のチェックボックスが選択されていることを確認する必要があります。

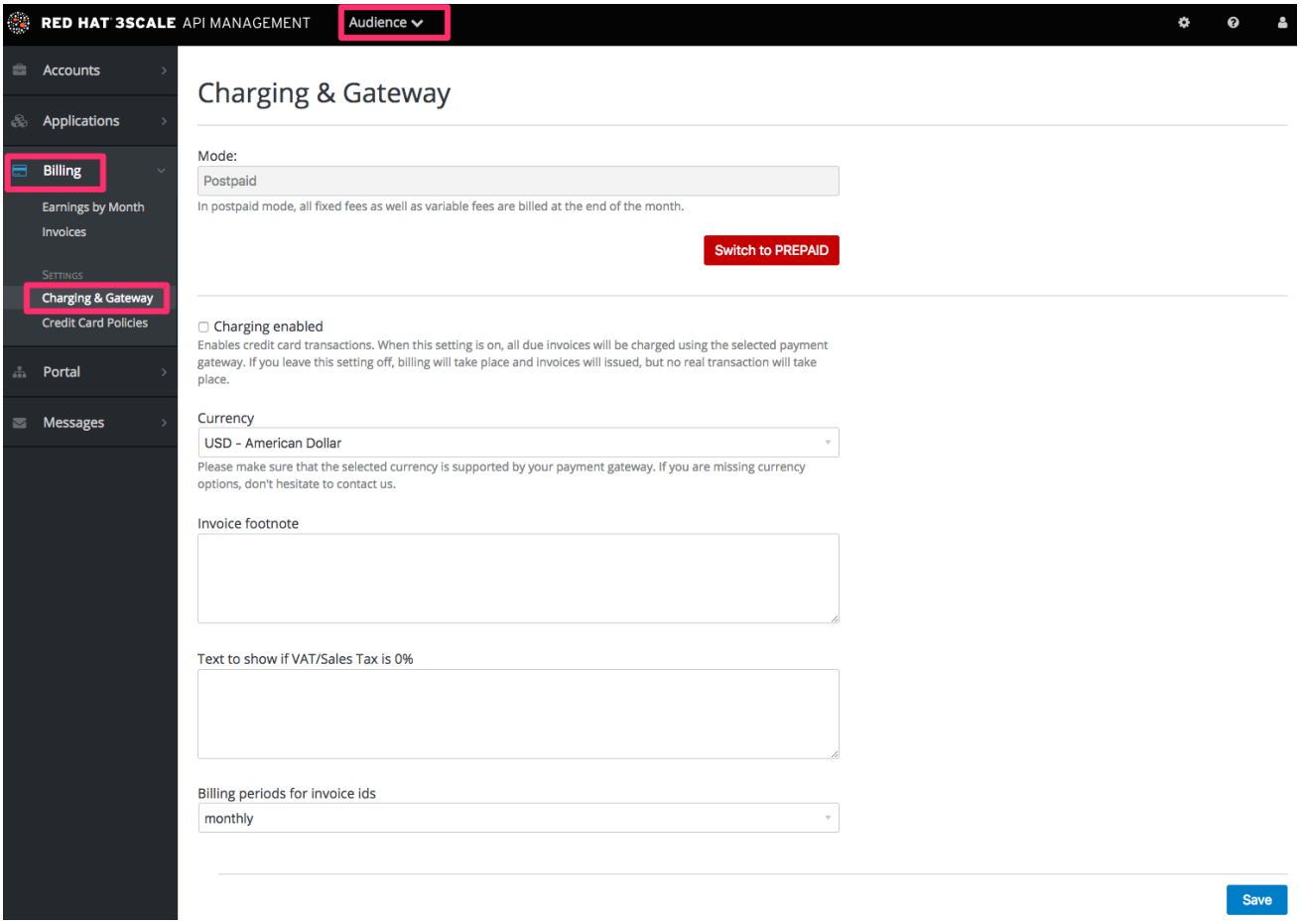


最後に、SHA-IN パスフレーズは、Date and origin verification ページにあります。

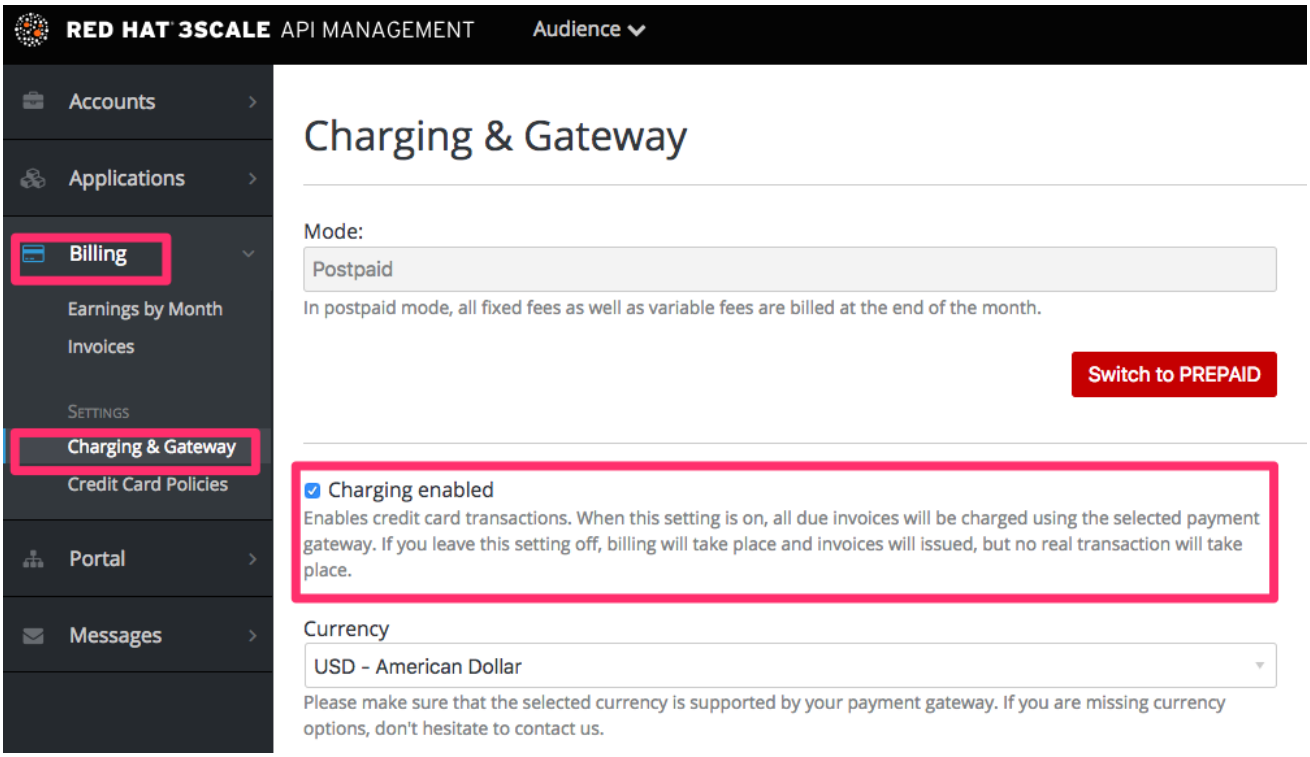


24.5.1.2.2. ステップ 2:3scale での設定

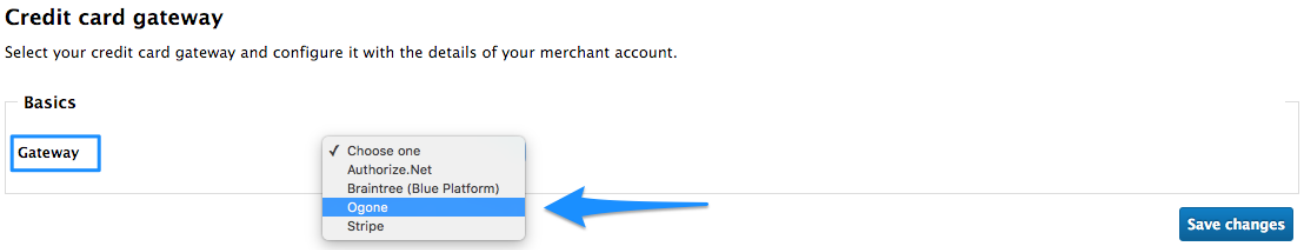
これらの API キーの使用を開始するように 3scale を設定する必要があります。そのためには、3scale 管理ポータルにログインし、**Settings** > **Billing** の順に移動します。



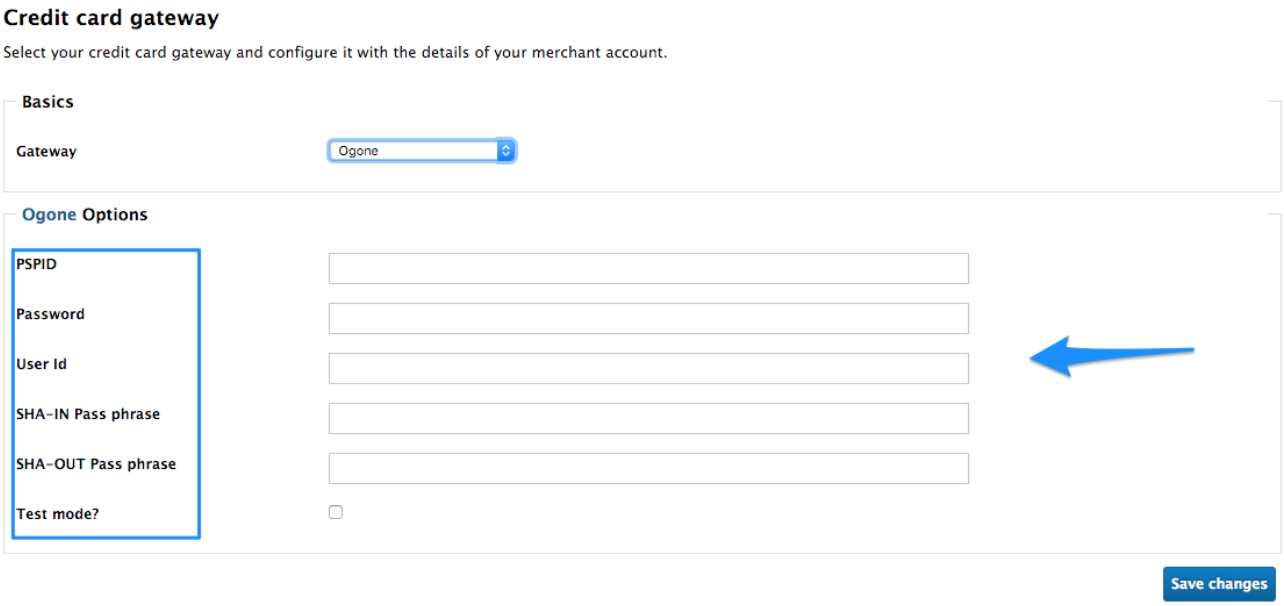
Charging enabled のチェックボックスが選択されていない場合には、選択して Save をクリックします。



ページ最下部近くの Gateway セクションに、ドロップダウンが表示されるはずです。これを Ogone に変更します。

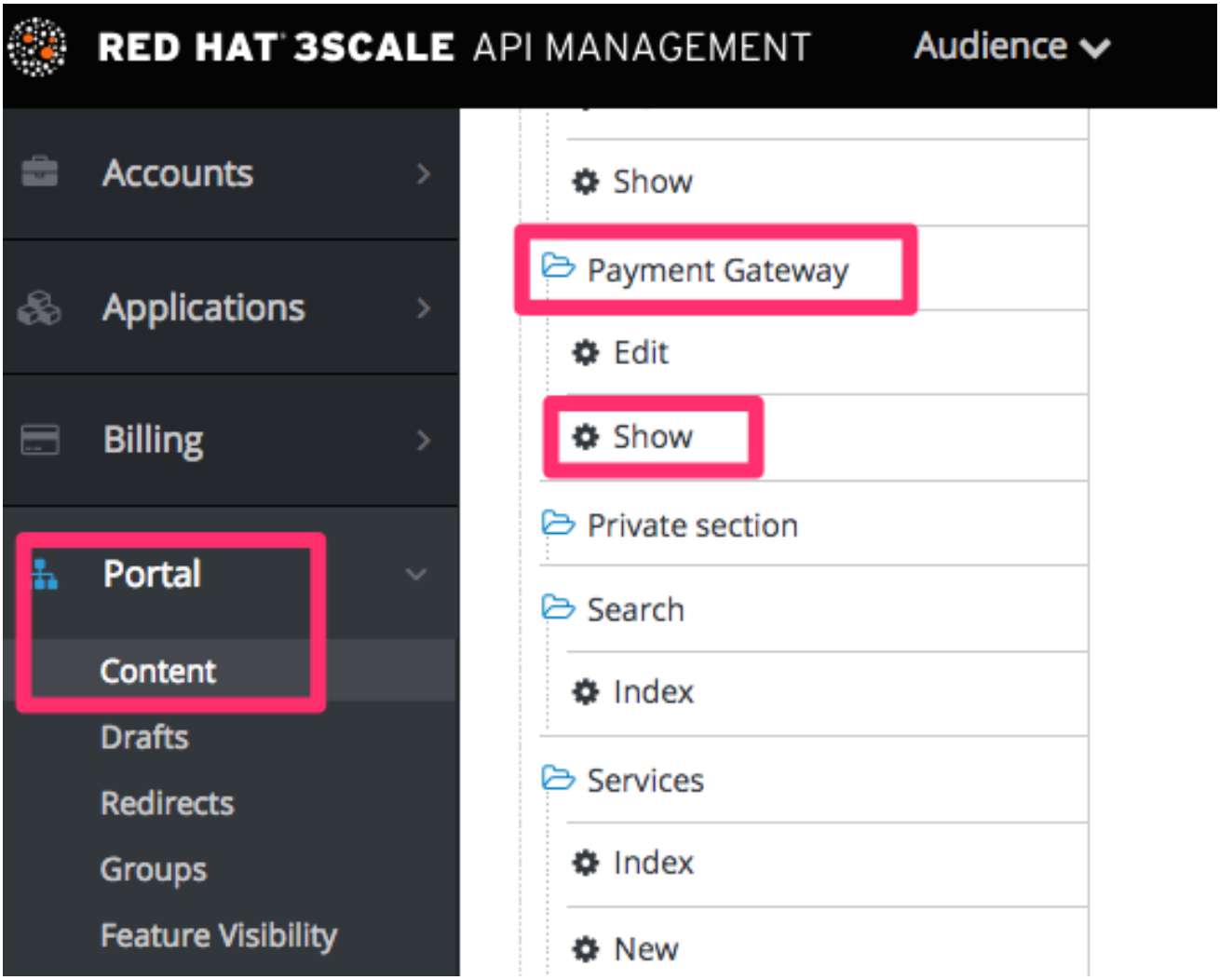


ドロップダウンの下のフォームが変わり、2 つのフィールドが表示されるはずです。Ogone の API キーを入力し、Save をクリックします。



支払いゲートウェイを変更すると、いくつかの警告が表示される場合があります。これは想定される状況です。表示されたら、警告を読んで了承してください。

これで支払いゲートウェイは設定されましたが、CMS では設定されていないため、ユーザーはまだ使用できない可能性があります。デベロッパーポータルのページに移動し、左側のナビゲーションペインでテンプレート Payment Gateway および Show をクリックします。



{% when "braintree_blue" %} の前に以下のコードがまだ表示されていない場合には、コード追加します。

```
{% when "ogone" %}
  {% if current_account.has_billing_address? %}
    {% if current_account.credit_card_stored? %}
      {% ogone_form "Edit Credit Card Details" %}
    {% else %}
      {% ogone_form "Add Credit Card Details" %}
    {% endif %}
  {% else %}
    <p><a href="{{ current_account.edit_ogone_billing_address_url }}">First add a billing address</a>
</p>
  {% endif %}
```

最後に、Save および Publish をクリックします。これで、ユーザーは Ogone ゲートウェイを使用して費用を支払うことができるはずです。

24.5.1.2.2.1. 注記

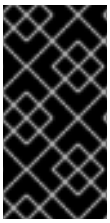
Ogone からのデータを 3scale のデータとマッピングするには、Ogone フィールド **alias** を使用することができます (設定は **3scale-[PROVIDER_ID]-[DEVELOPER_ACCOUNT_ID]**)。

24.5.1.2.2.2. トラブルシューティング

動作していない場合は、以下のヒントを確認してください。

- エイリアスマネージャーと連携するすべての支払い方法の概要は、次のドキュメントに記載されています。Payment methods processing/procedure overview.
- Ogone 管理コンソールで、Technical information → Transaction feedback に移動します。Directlink/Dynamic Parameters までスクロールし、すべてのパラメーターを選択します。これにより、エラーのトラブルシューティングが容易になります (エラーメッセージが請求書に含まれるため)。
- Ogone のパスワードを変更し、3scale コンソールでこの設定を更新してください。
- Special user for API (no access to the admin) チェックボックスを選択して、ご自分の Ogone アカウントにユーザーが作成されたかどうかを確認します。また、Ogone Support FAQ で API user を検索して詳細を確認することもできます。
- 3scale に、Ogone PSPID アカウントのパスワードではなく、API ユーザーのパスワードを入力するようにしてください。
- Ogone でサンドボックスモードを使用してエラーが発生した場合は、プロダクションに変更します。

24.5.1.3. Authorize.Net インテグレーション



重要

authorize.Net インテグレーションは非推奨になりました。新たなインテグレーションはサポートされません。2018 年 7 月 27 日より前に存在していたインテグレーションについては、Red Hat ではサポートを提供しますが、完全にサポートされる支払いゲートウェイのいずれかに移行することを推奨します。Stripe および Braintree。

Authorize.Net 支払いゲートウェイを Red Hat 3scale の請求システムと統合します。

前提条件:

- 有効な Authorize.Net アカウント
- 以下の Authorize.Net クレデンシャル
  - API ログイン ID
  - トランザクションキー

3scale API Management を設定します。

1. 3scale 管理ポータルにログインします。
2. **Settings → Billing** ページに移動します。
3. **Basics** セクションで、ドロップダウンメニューから **Authorize.Net** を選択します。
4. **Authorize.Net Options** セクションで、Authorize.Net API ログイン ID およびトランザクションキーを入力します。
5. **Save Changes** ボタンをクリックします。

第25章 課金設定

本セクションでは、API の使用に対して開発者に課金を行うさまざまな方法について説明します。

開設費

サービスのサブスクリプションに適用される 1 回限りの課金で、別のプランに切り替える場合には適用されません。サブスクリプションの初回月にのみ請求書/クレジットカードに現れます。アプリケーションプラン、サービスプラン、およびアカウントプランに設定することができます。

月額費用

毎月、繰り返し課金される費用です。サブスクリプションが月の途中で発生した場合には、日割り計算されます。固定費と呼ばれることもあります。アプリケーションプラン、サービスプラン、およびアカウントプランに設定することができます。

変動費

アプリケーションプランに設定された各メソッド/メトリクスに適用される課金ルールにより生じる費用です。この費用は API の使用量に基づくため、予め把握することはできず、請求期間が終了して初めて確定します。アプリケーションプランにのみ利用可能です。

サービスプランおよびアカウントプランは、SaaS 版 3scale のエンタープライズプランでのみ利用可能です。

例

> 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. 課金ルール

課金ルールにより、それぞれの API リクエストの費用を定義します。同じメトリクスに複数の課金ルールを設定した場合には、課金ルールが適用される範囲が分割されます。課金ルールは暦月に基づき、カウンターは各月の初日の 00:00 UTC にリセットされます。

例 1

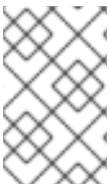
> Until 100 calls per month (from 1 to 100) each API call can be charged at $0.04, and starting from the call 101 (from 101 to infinity) the calls are charged at $0.10.

例 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.

例 3

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



注記

課金ルールは、メトリクスおよびメソッドに対して定義されます。実際の API リクエストは、マッピングルールを介してこれらのメトリクスおよびメソッドにマッピングされます。

25.2. 課金ルールの設定

1. **[Your_API_service] > Applications > Application Plans**の順に移動します。
2. 既存のアプリケーションプランを選択するか、新たなプランを作成します。
3. **Metrics, Methods, Limits & Pricing Rules**セクションで、**Pricing (x)** をクリックして課金セクションを開きます。
4. new pricing rule をクリックします。
5. From、To、および Cost per Unit の値を設定し、**Create pricing rule**をクリックします。

最後の 2 つのステップを繰り返し、必要なすべての課金ルール範囲を作成します。

ルールをそれ以降すべてに設定するには、To フィールドを空欄のままにします。

費用のメトリクスの小数部の最大桁数は 4 桁に設定されています。数値が 4 桁を超える場合には、4 桁の数字に丸められます。

25.3. 既存課金ルールの更新

1. edit をクリックします。
2. From、To、および Cost per Unit フィールドに関して、必要な修正を行います。
3. Update pricing rule をクリックします。

第26章 請求書

開発者アカウントがいずれかの有料サービスにサブスクライブしている場合には、請求プロセスによりそのアカウントに対して請求書が作成されます。設定された支払いゲートウェイを使用して、請求書に対する課金が行われます。

26.1. 請求書の一覧表示

Audience > Billing > Earning by monthセクションに、期限を迎えた請求書および受領済みの支払いの概要が表示されます。その月に発行した請求書のステータスごとに、すべての請求書の合計値が計算されます。

Total

キャンセル分を除くすべての請求書

In process

未確定、確定済み、および支払い待ちの請求書

Overdue

回収不能および課金に失敗した請求書

Paid

支払い済みの請求書

特定の月をクリックすると、その月の請求書をすべて一覧表示することができます。

Audience > Billing > Invoicesの順に移動して、請求書のリストを表示することもできます。ここでは、ID (平易な ID)、アカウント名、月、およびステータスで請求書を絞り込むことができます。

特定開発者アカウントの請求書は、アカウントの詳細ページ (**Audience > Accounts > Listing > [selected account]**) に表示されるパンくずリストの **X Invoices** から表示することもできます。ここで、X はそのアカウントの請求書数です。

26.2. 請求書ビュー

請求書リストで請求書の平易な ID をクリックすると、その請求書の詳細が表示されます。

26.2.1. 請求書の詳細

タイトルには、請求書の月および年、ならびに請求書が手動または自動請求プロセスで作成されたかが表示されます (例: **Invoice for February 2018 (manually created)**)。

その下の **Details** ブロックには、請求書の平易な ID、請求書のステータス、請求書の確定/発行日、ならびに支払い期限および支払い日が表示されます。PDF ファイルがすでに生成されている場合には、PDF ファイルをダウンロードするためのリンクも表示されます。

Issued by および **Issued to** ブロックには、それぞれ API プロバイダーおよび開発者アカウントの名前および住所が表示されます。

その下は **費用項目** が含まれるブロックで、計算された VAT/消費税、合計費用、および Text to show if VAT/Sales Tax is 0% のテキスト (設定されている場合) が表示されます。

請求書の **Transactions** ブロックには、支払いのステータス、そのタイムスタンプ、参照番号、支払いゲートウェイからのメッセージ、および金額など、クレジットカードによる支払いの試みがすべてリストとして表示されます。参照番号を使用して、支払いゲートウェイの管理コンソールで支払いを探すこ

とができます。

26.2.2. 請求書の編集

請求書が未確定または確定済みのステータスの場合には、請求書の詳細の一部を編集することができます。

請求書ビューの右上隅にある Edit のリンクをクリックすると、平易な ID および請求期間を変更することができます。

請求書の費用項目を追加および削除することもできます (合計、VAT/消費税、VAT/消費税込合計などの計算された費用項目を除く)。新たな費用項目を追加するには、Add をクリックして Name、Quantity (費用には影響を与えない)、Description、および Cost を入力します。

請求書ビュー最下部にあるリンクからは、請求書に対するアクションを実行することができます。請求書の現在のステータスに応じて、異なるアクションを実行することができます (例: PDF ファイルの生成、請求書の発行、請求書のキャンセル、PDF ファイルの再生成、課金、支払い済みの識別)。これらのアクション (PDF 関連を除く) により、請求書のステータスが変わります。

26.3. 請求書のステータス

請求書のステータスは、以下のいずれかです。

Open
: 請求書は作成されたが、自動計算はまだ完了していない。

Finalized
: 現在の請求期間の課金項目がすべて請求書に反映されている。

Pending
: 請求書は開発者に発行され、支払い待ちである。

Unpaid
: 請求額に対する課金に失敗したが、自動リトライ待ちである。

Paid
: 請求額に対する課金が正常に行われている。

Failed
: 請求額に対する課金に失敗し、これ以上自動リトライは行われない。

Cancelled
: 管理者によってキャンセルされている。

26.4. 自動請求プロセス

3scale では、請求プロセスは毎日実施されます。請求書を生成し、請求フローに従ってステータスを変更し、設定された支払いゲートウェイを使用して課金処理を実施します。

前払いモードと後払いモードでは、請求フローにわずかな違いがあります。3scale における請求は暦月に基づいているため、月の初日に特別なイベントが発生します。

26.4.1. 各月の初日

後払い

- **前** 月の変動費を請求します。費用は、未決済の請求書の費用項目として追加されます。
- **前** 月の未確定請求書を確定する。
- **現在の** 月の固定費を請求します。**未決済** 状態で今月の新しい請求書を作成します。

前払い

- 固定費を請求する (**当** 月)。
- 変動費を請求する (**前** 月)。

月の初めに確定した請求書に関する通知が API 管理者に送信されるので、管理者は請求書を確認して必要な調整を行うことができます。

上記の処理に加えて、日々行われるすべての操作も、月の初日に発生します。

26.4.2. 毎日

- 有効期限が切れたトライアルとまだ請求されていない新しい契約を請求する。今月の **未決済** 状態の請求書が作成されます。
- **前払い** のみ:未決済のすべての請求書を完了させます。ステータスが **完了** に変わります。
- 請求書を発行します。ステータスが **保留** に変わります。
  - 通常、請求書は確定してから 2 - 3 日後に発行されます。請求書の**Issued On**の日付はその日に設定され、**Due On**の日付 (請求書に対する課金が行われる日) は **Issued On** の 2 日後に設定されます。
  - 請求書が開発者に発行されると、開発者はメール通知を受け取り、発行された請求書をデベロッパーポータルで確認することができます。
- 請求書に対する課金を行う。
  - **Due On** の日付がその日またはそれ以前で、ステータスが **Unpaid** および **Pending** の請求書に対する課金が行われます。
  - 支払いが完了しなかった場合には、請求書のステータスは **Unpaid** に変わります。3 日後に課金がリトライされます。リトライに 3 回失敗すると、請求書のステータスは **Failed** に変わり、それ以上課金はリトライされなくなります。
- クレジットカードの有効期限切れに関する通知を行う。
  - 間もなくクレジットカードの有効期限が切れる開発者アカウントに、メール通知が送信されます。

26.4.3. 自動および手動の請求書生成

自動請求プロセスによって生成された請求書のヘッダーには、**(automatically created)** のラベルが付きます。以下に例を示します。**2019 年 1 月の請求書 (自動作成)**

手動で生成した請求書は、請求書の詳細ページで **(manually created)** と識別されます。

自動請求プロセスでは、当月 **open** のステータスにある既存の請求書を使用して、費用項目を追加する場合があります。ただし、その対象となるのは自動的に作成された請求書だけです。手動で作成した請求書は、自動請求プロセスにより更新されることはありません。

26.4.4. 期中のアップグレード

アプリケーション (またはアカウント/サービスサブスクリプション) が月の途中でアップグレードされた場合には、月額費用は残り日数に応じて日割り計算されます。アプリケーションプランで設定された制限はあん分されません。

アプリケーションが無料プランから有料プランにアップグレードされた場合には、次回の請求時に、日割り計算された月額費用を含む新しい請求書が生成されます。

アプリケーションが有料プランからより高額な有料プランにアップグレードされた場合には、さまざまな要素を加味して請求書が発行されます。

- Billing のモード:**前払い** または **後払い**
- いつプランが変更されたか

26.4.4.1. 前払い請求モード

1. アプリケーションプランが **請求日 (請求日は 8am UTC に始まります) に** 変更された場合には (**つまり、アプリケーションプランに対する請求がまだ行われていない**)、古いプランの固定費が請求書に記載され**払い戻し**の費用項目でディスカウントされます。新しいプランの固定費も請求書に追加されます。
   以下に例を示します。顧客が月の初日にプラン A ($200) にサインアップし、**同じ日に** プラン B ($300) にアップグレードした。この場合には、以下の費用項目が含まれる 1 通の請求書が生成されます。

説明	費用
固定費 (プラン A)	200
払い戻し (プラン A)	-200
アプリケーションのアップグレード (プラン A からプラン B へ)	300
Total	300

   顧客が月の途中でサインアップした場合には、固定費の 200 および 払い戻しの 200 は日割り計算される点に注意してください。

2. アプリケーションの **請求書がすでに発行された後に** アプリケーションプランが変更された場合。

   - **アップグレード** の場合には、開発者には 2 通の請求書が発行されます。1 つは当初請求用で、もう 1 つはアップグレード用です。
     以下に例を示します。顧客が月の初日にプラン A ($200) にサインアップした後に、月の途中でプラン B ($300) にアップグレードした。この場合には、以下のような請求書が生成されます。

説明	費用
固定費 (プラン A)	200

説明	費用
Total	200

説明	費用
払い戻し (プラン A)	-100
アプリケーションのアップグレード (プラン A からプラン B へ)	150
Total	50

請求期間の途中でアップグレードが行われたので、2 通目の請求書では、払い戻された費用 ($100) および新たな費用 ($150) は日割り計算されます。

- アプリケーションの **ダウングレード** (より低額なプランへの変更) に伴う払い戻しは、現在サポートされていません。

26.4.4.2. 後払い請求モード

後払い請求モードでは、**払い戻し** および **アプリケーションのアップグレード** に関する費用項目が含まれる 1 通の請求書が発行されます。

重要:この動作は、以下の変更と共に **2018 年 4 月 20 日**に導入されました。

- アプリケーションが作成と同じ日にアップグレードされた場合に、当初請求額 (元のアプリケーションプラン用) が請求書に追加されない、というバグが修正されました。
- 従来は、アプリケーションのアップグレード時には、新旧プランの差額に関する 1 つの費用項目しか追加されませんでした。たとえば、上記 **前払い請求モード** セクションで説明したシナリオ 2 では ($200 のプラン A から $300 のプラン B へのアップグレード)、2 番目に生成される請求書は以下のようになりました。

説明	費用
アプリケーションのアップグレード (プラン A からプラン B へ)	50
Total	50

ここで、$50 は月の残りを考慮して日割り計算した新旧プランの差額です ($150 - $100)。

2018 年 4 月 20 日以降は、合計費用は以前と同じままですが、請求書の計算がより明確に反映され、払い戻しと課金が別々に追加されるようになりました。

26.5. アカウントごとの請求/課金の有効化/無効化

自動請求プロセスでは、有料サービスにサブスクライブしているすべての開発者アカウントの請求書が生成されます。

請求はグローバルに有効または無効にすることができます。請求についての詳細は、請求設定の定義 を参照してください。ただし、開発者アカウントごとに、請求および課金の両方を有効または無効にすることもできます。これを行うには、アカウントページに移動し、対応するボタン (有効化/無効化) をクリックします。

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

課金を無効にして請求を有効にした場合には、開発者には請求書は発行されますが、課金は行われません。これは、課金を別個に処理する場合 (送金など) に便利です。

両方を無効にすると、開発者には請求書は発行されず、課金も行われません。

請求を無効にして課金を有効にした場合には、開発者には新たな請求書は発行されませんが、支払いの済んでいない請求書 (**Pending** および **Failed**) は引き続き課金されます。

この **Billing Status** ブロックで、アカウントでクレジットカード情報が設定されているかどうかを確認することもできます。クレジットカード情報が設定されている場合には、カードの有効期限の月および年が表示されます。クレジットカード情報の有無により、サインアップおよびプラン変更時のデベロッパーポータルのフローが異なる場合があります (詳細については、「クレジットカードに関する作業フロー」を参照してください)。

26.6. トライアル期間が設定されたプラン

有料プランにトライアル期間を設定し (日数単位)、指定した日数だけ課金を遅らせることができます。

アプリケーションのトライアル期間のステータスは、アプリケーションリスト (**[Your_product_name] > Applications > Listing**) の **State** 列 (**live – trial expires in N days**と表示) およびアプリケーションの詳細ページの **Trial days left** フィールドに表示されます。

トライアル期間中、アプリケーションではすべての機能およびプランで定義した使用制限を使用することができます (制約はありません)。

一度トライアル期間を設定したら、トライアルをキャンセルしたり期間を延長したりすることはできません。別のプランにアップグレードしても、トライアル日数のカウンターはリセットされません。

トライアル期間が終了する 10 日前に、間もなくトライアル期間が終了することを知らせるメール通知が開発者アカウントに送信されます。このメールのテンプレートは、**Audience > Messages > Email Templates** で設定することができます。アプリケーションプランの場合は **"Application trial period expired for buyer"**、トライアル期間のあるサービスプランの場合は **"Service trial period expired for buyer"**。開発者は、現在のプランをそのまま使用するか、別のプランに切り替えるか、またはサブスクリプションをキャンセルする場合には API プロバイダーに連絡する必要があります。

トライアル期間が終了すると、次回自動請求が実行される時に (「自動請求プロセス」を参照) 開発者に課金が行われます。この課金は、現在アプリケーションまたはサービスに設定されているプランに規定された費用に基づきます。トライアル期間が終了した後もアプリケーションがブロックまたは一時中断**されることはなく**、引き続き機能する点に注意してください。

技術的には無料プランにトライアル期間を設定することはできますが、機能の有用性が失われるため推奨されません。

26.7. VAT レート/消費税

付加価値税 (VAT) は、欧州連合内で販売される商品およびサービスに適用される税です。他の国には、消費税などの類似の税があります。API サービスの使用に対して顧客に課金する API プロバイダーは、既存の規則に準拠するために、通常は税金を請求してそれを請求書に反映する必要があります。

3scale の請求では、設定により VAT を自動的に請求書に含めることができます。

26.7.1. VAT rate フィールドの設定

VAT レートは、請求書の合計費用に適用される数値です (パーセント単位)。VAT レートを設定するには、管理ポータルの Dashboard から以下の手順に従います。

1. **Audience > Accounts > Fields Definitions**の順に移動します。
2. **Account** セクションで **Create** をクリックします。
3. [new field] のドロップダウンリストで **vat_rate** を選択します。
4. **Label** の値を設定します。
5. チェックボックスを設定して、開発者に対してフィールドを **Required (必須)** とするか、**Read only (読み取り専用)** とするか、または **Hidden (非表示)** とするかを定義します。
   注記:Choices フィールドに入力することができるのは文字列だけです。したがって、このフィールドを使用することはできません。
6. **Create** をクリックします。

26.7.2. VAT code フィールドの設定

VAT コードは、VAT の識別番号です。VAT コードを設定するには、管理ポータルの Dashboard から以下の手順に従います。

1. **Audience > Accounts > Fields Definitions**の順に移動します。
2. **Account** セクションで **Create** をクリックします。
3. [new field] のドロップダウンリストで **vat_code** を選択します。
4. **Label** の値を設定します。
5. チェックボックスを設定して、開発者に対してフィールドを **Required (必須)** とするか、**Read only (読み取り専用)** とするか、または **Hidden (非表示)** とするかを定義します。
6. **Create** をクリックします。

26.7.3. アカウントの VAT 値の設定

VAT code および VAT rate フィールドを追加したら、開発者アカウントの Edit フォームで対応する値を設定することができます。フィールド定義で選択したチェックボックスに応じて、開発者はデベロッパーポータルでフィールドを表示したり編集したりすることもできます。

VAT コードには任意の文字列を使用することができます。

VAT レートは数値でなければなりません。整数または 10 進数のいずれかを使用でき (例:21、23.5 など)、VAT として含まれる費用のパーセンテージを表します。

26.7.4. VAT が含まれる請求書

ゼロではない VAT レートが設定された開発者アカウントに対して請求書が生成されると、請求書には以下の費用項目が含まれます。

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

<VAT rate> は、VAT rate フィールドで設定した Label の値に置き換えられます。**<VAT rate value>** は、請求書が発行されるアカウントに設定した値です。

VAT コードは、VAT コードのフィールド定義で Label として指定したラベルと共に、請求書の PDF バージョンに追加されます。値は、開発者アカウントの詳細で設定された値に対応します。

フィールド **vat_rate** および **vat_code** は、3scale Account Management API からでも読み取りおよび書き込みが可能です。

26.8. デベロッパーポータルビュー

開発者は、デベロッパーポータルでクレジットカード情報を管理し、請求書を表示することができます。

開発者がデベロッパーポータルで請求に関する要素を確認できるためには、Finance スイッチ (**Audience > Developer Portal > Feature Visibility**) が **Visible** のステータスでなければなりません。

ログインしているユーザーは、**Settings > Credit Card Details** でクレジットカード情報を更新することができます。設定されている支払いゲートウェイによってフォームが異なる場合や、初めに請求先住所を入力しなければならない場合があります。

注記

クレジットカード情報は、3scale のデータベースには保管されません。これらの情報は、支払いゲートウェイによって管理されます。3scale データベースには、支払いゲートウェイから提供されたクレジットカード番号の最後の 4 桁、有効期限、および参照のみが保存されます。

デベロッパーポータルでは、開発者は 1 つのクレジットカードしか使用することができません。

クレジットカード情報をデベロッパーポータルから削除することはできません。開発者アカウントのユーザーがクレジットカード情報の削除を希望する場合には、API プロバイダーにクレジットカード情報をシステムから削除するよう要求する必要があります。

請求書を表示するには、**Settings > Invoices** の順に移動します。そこで各請求書の詳細を表示するか、請求書の PDF ファイルをダウンロードすることができます。

26.9. クレジットカードに関する作業フロー

26.9.1. 有料プランへのサインアップ

開発者が有料プランにサインアップすると、アプリケーションのクレデンシャルを表示するために、クレジットカード情報の入力が要求されます。開発者が初めてデベロッパーポータルにログインすると、Credit Card Details のページにリダイレクトされます。開発者アカウントの他のページにアクセスを試みても、再び Credit Card Details のページにリダイレクトされます。

2016 年 7 月 5 日より前に 3scale アカウントが作成された場合、**Audience > Developer Portal > Feature Visibility** で Credit Card on Signup のスイッチを設定して、この動作を手動で有効にする必要があります。それ以外の場合は、クレジットカード情報が設定されていない場合でも、アプリケーションのクレデンシャルが表示されます。

Liquid タグを使用してすべてのメニュー項目を非表示にすることができます。ただし、該当するデベロッパーポータルテンプレートをカスタマイズして、**Credit Card Details** タブを除外することができます。

current_account Liquid drop はメソッド **requires_credit_card_now?** を公開します。このメソッドは、クレジットカード情報が登録されていないが必要な場合 (管理ポータルで Billing が設定されている場合のみ) には **true** を返し、それ以外の場合には **false** を返します。

submenu および **users_menu** パートのメニュー項目およびその他のユーザーインターフェイス要素を、非表示にすることができます。そのためには、それらを以下の条件でラップします。

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

26.9.2. 無料プランから有料プランへのアップグレード

既存アプリケーションのプラン変更については、複数のオプションを設定することができます (例: 開発者が直接プランを変更する、開発者がプランの変更を要求する等)。アプリケーションを無料プランから有料プランにアップグレードする場合には、アップグレードの前に開発者がクレジットカード情報を入力していることが重要です。カード情報の入力方法は、**[Your_product_name] > Integration > Settings** の **APPLICATION PLAN CHANGING**セクションで、以下のオプションにより設定することができます。

開発者がクレジットカードを所有している場合には、直接プランを変更します。それ以外の場合には、
・プラン変更の要求だけをする
・有料プラン用にクレジットカード情報の入力を要求する

開発者が変更を要求することだけを許可する場合には、最初のオプションを選択してください。この場合、クレジットカード情報が入力された後に手動でアップグレードを実行します。

有料プランにアップグレードするにはクレジットカード情報を入力する必要があることを開発者に通知する場合には、2 番目のオプションを選択してください。プラン変更のウィジェットで開発者に You cannot change plan until you enter your Credit Card details というメッセージが表示され、クレジットカード情報のフォームがポイントされます。

26.10. 請求先住所フィールド

開発者がデベロッパーポータルの Credit Card Details のページで請求先住所を入力する際、この住所は法人アカウント住所とは別に保管されます。

デフォルトでは、請求書の **Issued to** フィールドには法人住所が表示されます。ただし、開発者が法人住所を設定しておらず、請求先住所しか設定していない場合には、請求書には後者が使用されます。この場合、組織名は住所の一部と見なされます。

デフォルトでは、請求先住所は管理ポータルまたはプロダクト API では表示されません。ただし、以下の手順により追加することができます。

1. **Audience > Accounts > Fields Definitions**の順に移動します。

2. **Account** ブロックで **Create** をクリックします。

3. ドロップダウンリストから billing_address を選択し、ラベルを追加して Read only のチェックボックスを選択し、**Create** をクリックします。

これで、Account Management API および Webhook の Accounts の XML モデルに、**<billing_address>** フィールドが表示されるようになります。対応する請求先住所が、管理ポータルの Account および Edit account のページで読み取り専用として表示されます。

開発者は、デベロッパーポータルの Credit Card Details セクションで、請求先住所を変更することができます。管理者は管理ポータルで請求先住所を変更することはできませんが、以下のフィールドを使用して (追加フィールドとして)、Account Management API の **Account Edit** エンドポイントにより変更することができます。

```
billing_address_name
billing_address_address1
billing_address_address2
billing_address_country
billing_address_city
billing_address_state
billing_address_zip
billing_address_phone
```

第27章 メール通知

請求に関するさまざまなイベントがトリガーとなり、API プロバイダーおよび開発者にメール通知が送信されます。

27.1. プロバイダーへの通知

3scale アカウントのユーザー (管理者および **Billing** 権限を持つメンバー) は、請求に関する通知をサブスクライブおよびサブスクライブを解除することができます。その設定は、**Account Settings (右上の歯車アイコン) > Personal > Notification Preferences** の Billing セクションで行います。

Action required: review invoices

請求サイクル最終日の数日前に送信されるので、顧客に送信する前に請求書を確認することができます。

Customer downgraded

顧客がより低額の月額固定プランに変更した際に送信されます。

Expiring credit card

間もなく顧客のクレジットカードの有効期限が切れる場合に送信されます。

Payment error (retry)

課金に失敗し、請求書のステータスが Unpaid となりリトライ対象となった場合に送信されます。

Payment error (final)

課金の最後のリトライに失敗し、請求書のステータスが Failed となった場合に送信されます。

注記:3scale アカウントのすべての管理ユーザーは、請求に関する通知をサブスクライブしている場合、それらの通知を受け取ります。

27.2. 開発者へのメール通知

開発者アカウントに送信されるメール通知は、**Audience > Messages > Email Templates**で設定することができます。以下のメール通知を利用することができます。

Credit card expired notification for buyer

間もなくクレジットカードの有効期限が切れる場合に送信されます。

Invoice charged successfully for buyer

請求書に対する課金が正常に完了した場合に送信されます。

Invoice charge failure for buyer with retry

請求書に対する課金に失敗し、請求書が Unpaid のステータスとなり、課金のリトライが行われる場合に送信されます。

Invoice charge failure for buyer without retry

請求書に対する 3 回目の課金に失敗し、請求書が Failed のステータスとなり、それ以上課金のリトライは行われません。

Upcoming invoice charge for buyer

請求書が開発者に発行された場合に送信されます。

開発者アカウントのすべての管理ユーザーは、上記の通知を受け取ります。

27.2.1. 請求に関するメールアドレス

請求に関する問題を解決するために顧客が連絡することのできるメールアドレスを設定することができます (例: billing@example.com)。その設定は、**Audience > Messages > Support Emails**の **Finance support email** フィールドで行います。

電子メールのテンプレートは、Liquid drop **{{ provider.finance_support_email }}** を使用してメールアドレスを参照します。

第28章 BILLING API

Billing API により、共通的な請求プロセスを自動化することができます。

Billing API のすべてのエンドポイントは、管理ポータルの **Documentation (?) > 3scale API Docs > Billing API** で確認することができます。

Billing API には、以下の要件を満たす有効なアクセストークンが必要です。

- プロバイダーアカウントの管理ユーザーまたは Billing 権限を持つメンバーユーザーのいずれかに属していること
- Billing API スコープが含まれていること

パラメーターとして請求書 ID が必要な場合には、平易な請求書 ID **ではなく** 請求書 ID を指す点に注意してください。

API エンドポイントの XML レスポンスは見て直ぐに理解でき、請求書のフィールドは Web および PDF 版の情報と同じ内容を示しています。

注意を要するレスポンスのフィールド

- **creation_type**:**manual**(手動で作成された請求書の場合) または**background**(3scale の自動請求プロセスにより作成された請求書の場合) どちらかの値をとります。
- **provider**: API プロバイダー (管理アカウント) の詳細で、請求書の **Issued by** セクションに対応します。
- **buyer**: 開発者アカウントの詳細で、請求書の **Issued to** セクションに対応します。

XML 版の請求書には、**line-items** フィールドに費用項目のリストも含まれます。

一部の費用項目 (特に自動的に作成された項目) では、一般的な名前、説明、量、および費用 (価格) 以外に、以下の項目が表示されます。

- **type**: 費用項目のタイプで、以下の値を取ります。
  - **LineItem::PlanCost**: 定額プランの費用項目の場合
  - **LineItem::VariableCost**: 変動費の費用項目の場合
- **metric_id**: 変動費の費用項目用で、費用が関連付けられているメトリクスの ID
- **contract_id**: 費用が関連付けられているサービスまたはアプリケーションの ID

パート V. 解析

第29章 API アクセスを管理および最適化するための 3SCALE API 解析の実装

3scale API 解析を実装して API アクセスを管理および最適化することで、時間と共に変化する使用状況の傾向などを追跡することができます。トラフィックを管理し、ピーク時に対応し、API に最も多くのリクエストを送信しているユーザーを特定するためには、ご自分の API がどのように使用されているかを知ることが重要なステップとなります。

3scale では、以下のレベルで定義可能なメソッドおよびメトリクスに関する API 解析を収集します。

- プロダクト:**Hits** は、API へのトラフィックを追跡する組み込みのメトリクスです。追加のメトリクスを作成して、解析を取得する API にメソッドを指定することができます。
- バックエンド:3scale では、メソッドおよびメトリクスが API バックエンドに登録されます。これにより、メソッドおよびメトリクスが API バックエンドを使用する各プロダクトに属するかのように機能します。バックエンドレベルのメトリクスの制限および課金ルールを、プロダクトレベルで定義されるアプリケーションプランに設定することができます。
- アプリケーション:3scale で作成された各アプリケーションの解析レポートを取得することができます。

前提条件

- スタートガイドの手順 を完了している。
  スタートガイド を使用して、既存の 3scale コードプラグインのいずれかを使用してインテグレーションを実施します。
- あるいは、他のインテグレーション方法の類似作業フローに従います。利用可能なインテグレーションオプションの詳細は、API ゲートウェイの管理の APIcast の運用 の章を参照してください。

29.1. API の使用状況を把握する 3SCALE API メトリクスおよびメソッド

3scale は、API プロダクトの統計値に関して無限にスケーラブルなデータリポジトリーとして機能します。メトリクスおよびメソッドを使用して API プロダクトの統計値を取得することで、API へのアクセスを最適に管理するために必要な情報を取得することができます。以下に例を示します。

- **Hits/transactions**: API プロダクトへの呼び出しの数。Hits は、すべての API にデフォルトでメトリクスとして含まれています。Hits は、API プロダクトへの全呼び出し数とすることも、API プロダクトの個々のメソッドにブレークダウンすることもできます。
- **Data transfer**: API プロダクトを通じてアップロードおよびダウンロードされたデータ量 (MB/GB 単位)
- **CPU hours**: API プロダクトへの呼び出しに関連する処理時間 (またはその他の内部リソース)
- **Results returned**: 返されているレコードまたはデータオブジェクト数のカウント
- **Disk storage**: アカウントにより使用されているディスクストレージの合計

ご自分の API プロダクトに関連するより多くのメトリクスを追跡することができます。時間の経過と共に増加する数量であれば、3scale では任意の数のメトリクスおよびメソッドを追加することができます。

使用するメトリクスを選択したら、プロダクトおよびバックエンドへのメトリクスの追加 に記載の手順に従って、管理ポータルにそれらのメトリクスを登録します。

選択したプロダクトまたはバックエンドにメトリクスおよびメソッドを追加することができます。それらに分かりやすい名前およびシステム名を設定します (3scale はこれらの名前をプラグイン設定で使用します)。メソッドおよびメトリクスの作成に関する詳細は、使用状況の詳細を把握するためのメソッドの指定およびメトリクスの追加 を参照してください。

29.2. API メトリクスを取得するための 3SCALE プラグインの設定

3scale システムで追跡するメトリクスの名前を設定したら、それらのメトリクスを報告するようにプラグインを設定する必要があります。この操作の詳細な手順は、使用しているプラグインまたはインテグレーション手段によって異なります。デフォルトでは、プラグインは **Hits** (API トランザクション) メトリクスしか報告しません。

前提条件

- プロダクトまたはバックエンドにメトリクスがすでに設定されている必要があります。

手順

1. 着信 API コールで決められたとおりに、適切なメトリクス/メソッド名をプラグインに渡します。
   必要なメトリクス/メソッドの値および増分は、プラグインが公開する承認またはレポートメソッドの引数です。

2. メトリクスで **System name** メソッドを使用して、特定の API メソッドのトラフィックを報告します。
   これにより、報告されたメソッドと **Hits** メトリクスの両方のカウンターに、自動的に増分が加算されます。

3. 管理ポータルの **Account Settings** > **Integrate** → **3scale API Docs**で、**3scale Service Management API** を使用してトラフィックを報告します。

さまざまなエンドポイントに関する補足情報については、管理ポータルの **Account Settings** > **Integrate** → **3scale API Docs** から利用できる**3scale API Documentation** を参照してください。

29.3. 3SCALE API バックエンドの解析値の表示

バックエンドとして設定された特定の API のトラフィックデータを表示することができます。**Traffic** ページにエンドポイントへのヒットカウントが表示されます。

前提条件

- API バックエンドがすでに作成されている必要があります。プロダクトに追加するバックエンドの作成 を参照してください。

- バックエンドが API プロダクトに追加されている必要があります。既存プロダクトへのバックエンドの追加 を参照してください。

手順

1. 管理ポータルの Dashboard から、**[Your_backend_name]** > **Analytics** の順に移動します。

2. レポートの期間を選択します。過去 24 時間、7 日間、30 日間、または 12 カ月間の表示を選択することができます。また、期間を指定することもできます。

3. オプション:**Hits** 以外の解析値 (メソッドまたは他の最上位メトリクスのいずれか) を選択します。

   - 結果がチャートに表示されます。報告するトラフィックがない場合は、以下のメッセージが表示されます。**選択した期間のデータはありません**

4. **Download CSV** リンクをクリックして、データを CSV ファイルにダウンロードします。

29.4. API メトリクスを取得するためのプラグイン設定を確認するためのテストリクエストの送信

API と 3scale 間の接続を確立します。これにより、トラフィックを API プロダクトに送信し、それが API Analytics ダッシュボードに登録されるのを確認することができます。

前提条件

- 3scale の開発者アカウント
- 3scale アプリケーションおび API クレデンシャル

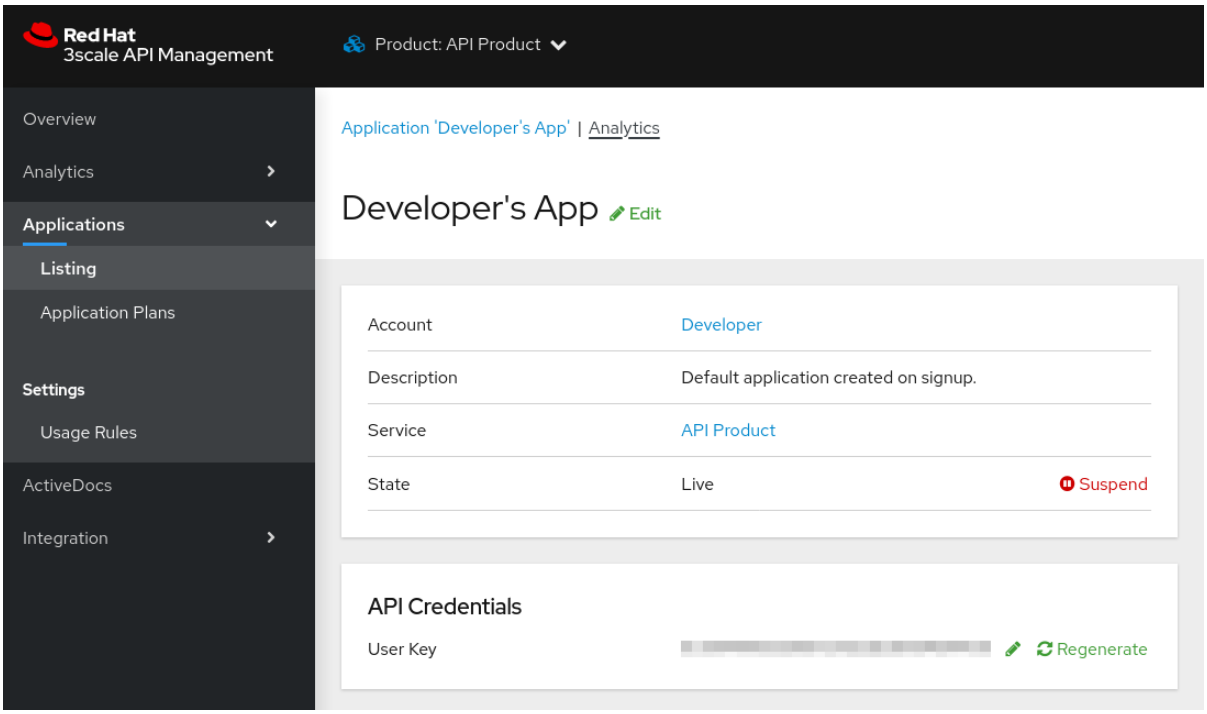
手順

1. **Audience > Applications > Listing**の順に移動し、既存アプリケーションのリストを表示します。

2. 名前をクリックしてアプリケーションを選択します。

3. 選択したアプリケーションの **API クレデンシャル** を確認します。クレデンシャルは選択した認証タイプによって異なり、ユーザーキー (API キー)、アプリケーション ID とアプリケーションキー、またはクライアント ID とクライアントシークレットのいずれかです。
   利用可能な認証モードの詳細は、認証パターン を参照してください。

図29.1 API クレデンシャル



4. これらのキーを使用して、通常の方法で API を呼び出します。たとえば、**cURL** を使用してコマンドラインから呼び出すか、**GET** メソッドを使用して API エンドポイントのブラウザーから呼び出します。呼び出しの詳細は、実際の API プロダクトのメソッド構造により異なります。これらの呼び出しからのトラフィックが、API プロダクトの Analytics セクションに表示されます。

次のステップ

デフォルトでは、API プロバイダーは管理ポータルを通じて、アプリケーションを作成した開発者はデベロッパーポータルを通じて、それぞれ使用状況の統計を表示することができます。それぞれの開発者は、自分自身のアプリケーションの使用状況の統計しか表示することはできません。デベロッパーポータルから解析ビューを非表示にして、表示できるユーザーをさらに制御することができます。デベロッパーポータルのカスタマイズに関する詳細は、デベロッパーポータルの作成 を参照してください。

Analytics セクションの使用状況グラフ以外にも、Analytics API を使用して解析データにアクセスすることができます。3scale Analytics API により、API プロダクトのすべての解析データを、XML または JSON いずれかの形式で柔軟に抽出することができます。

解析機能にアクセスするための他の方法には、以下が含まれます。

- **毎日および週ごとのトラフィックレポート**: これらのレポートはトラフィックに関する集計データを提供します。これには、ご自分の API プロダクトへの新規サブスクライバーおよび上位のアプリケーションに関する情報が含まれます。管理ポータルの **Account Settings (歯車アイコン) > Personal > Notification Preferences** でこれらのレポートを有効にするには、 **Daily report** および **Weekly report** チェックボックスを見つけます。有効にすると、3scale アカウントの管理ユーザーがこれらのレポートを電子メールで受信します。

- **CSV エクスポート**: 各解析ビューページには **download CSV** ボタンリンクがあり、.CSV フォーマットで使用状況の統計をダウンロードできます。

29.5. 3SCALE API の解析が表示されない場合のトラブルシューティングテクニック

トラフィックが **[Your_product_name] > Analytics > Traffic**の使用状況チャートに表示されない場合は、以下のチェックを行います。

- **承認/レポート呼び出しは正常に応答しているか？**
  すべてのプラグインは 3scale Service Management API を呼び出しますが、この API には事前に定義されたレスポンスコードがあります。有効なキーによる承認呼び出しは、HTTP コード **200** と共にレスポンスを返します。レポート呼び出しは、コード **202** と共に応答します。

- **インテグレーションエラーコンソールにエラーは表示されていないか？**
  3scale によって検出されたインテグレーションエラーのログは、**[your_product_name] > Analytics > Integration Errors** で確認することができます。

- **正しいメトリクスおよびメソッド名が使用されているか？**
  異常が発生する最も一般的な原因は、レポート呼び出しに渡されるメソッドおよびメトリクス名が、ご自分の管理ポータルの API 設定で作成される名前に対応していないことです。それぞれのメトリクス/メソッドについて、正しい **システム名** が使用されていることを確認してください。

- **マッピングルールは正しくメトリクスにマッピングされているか？**
  マッピングルールが正しくメトリクスにマッピングされていないと、データが Analytics に表示されない場合があります。API プロダクトおよび API バックエンドの両方について、マッピングルールが正しくメトリクスにマッピングされていることを確認します。

  - プロダクト:管理ポータル Dashboard から、**[Your_product_name] > Integration > Methods & Metrics** の順に移動します。

  - バックエンド:管理ポータル Dashboard から、**[Your_backend_name] > Methods & Metrics** の順に移動します。

第30章 組み込み機能を超える 3SCALE API 分析値のエクスポート

デフォルトの機能では提供されない情報を取得できるように、組み込まれた 3scale 分析の機能を拡張するスクリプトを作成します。

Account Management API および Analytics API を使用すると (Enterprise のみ)、必要な情報を希望の形式で取得するスクリプトを作成することができます。本項のユースケースを使用して、必要なデータを 3scale から取得する独自のシナリオに役立てることができます。

カスタムスクリプトが重要な理由

3scale では、API Dashboard で利用可能な機能を継続的に改善しています。しかし、お客様が当社の開発プランに先立ち、まだサポートされていない非常に特殊な機能を必要とする場合があります。

API 管理のニーズを満たすために、3scale 管理ポータルはすべてのデータにアクセスするのに必要なツールを提供します。スクリプトの作成にはリソースが必要ですが、カスタマイズしたスクリプトでは、さまざまなユースケースを実装するための完全な柔軟性と制御性が得られます。

30.1. 3SCALE を使用してアプリケーションの使用状況に関するデータを抽出する例

あるお客様は、毎週、数千人もの新規開発者を受け入れていました。3scale では必要な作業 (キーのプロビジョニング、サインアップワークフロー、電子メールによる連絡など) が自動化されているため、このお客様は受け入れプロセスにある程度対応することができていました。しかし、お客様にとって非常に重要だが、3scale では実施できないことがありました。

お客様は多数の開発者を受け入れていたため、運用およびマーケティングチームが新規開発者とより効果的に交流できるように、API の使用状況に応じて新規開発者を分類するためのシンプルな手段が必要でした。このような機能は、少なくとも求められるレベルの詳細さでは、3scale で提供される組み込み解析ツールではまだ利用することができませんでした。しかし、すべてのデータにはシステムからアクセスすることができたので、3scale の Account API および Analytics API を使用してそのデータを抽出することができました。

以下に例を示します。お客様の要件

お客様は、過去 10 日間に無料の評価版プランにサインアップした新規開発者の数を知り、それをいくつかのグループに分けたいと考えました。

まず、サインアップはしたがお客様の API を全く使用しなかった開発者の数を知りたいと考えました。

次に、API を使用した開発者を以下の 2 つのグループに分けたいと考えました。

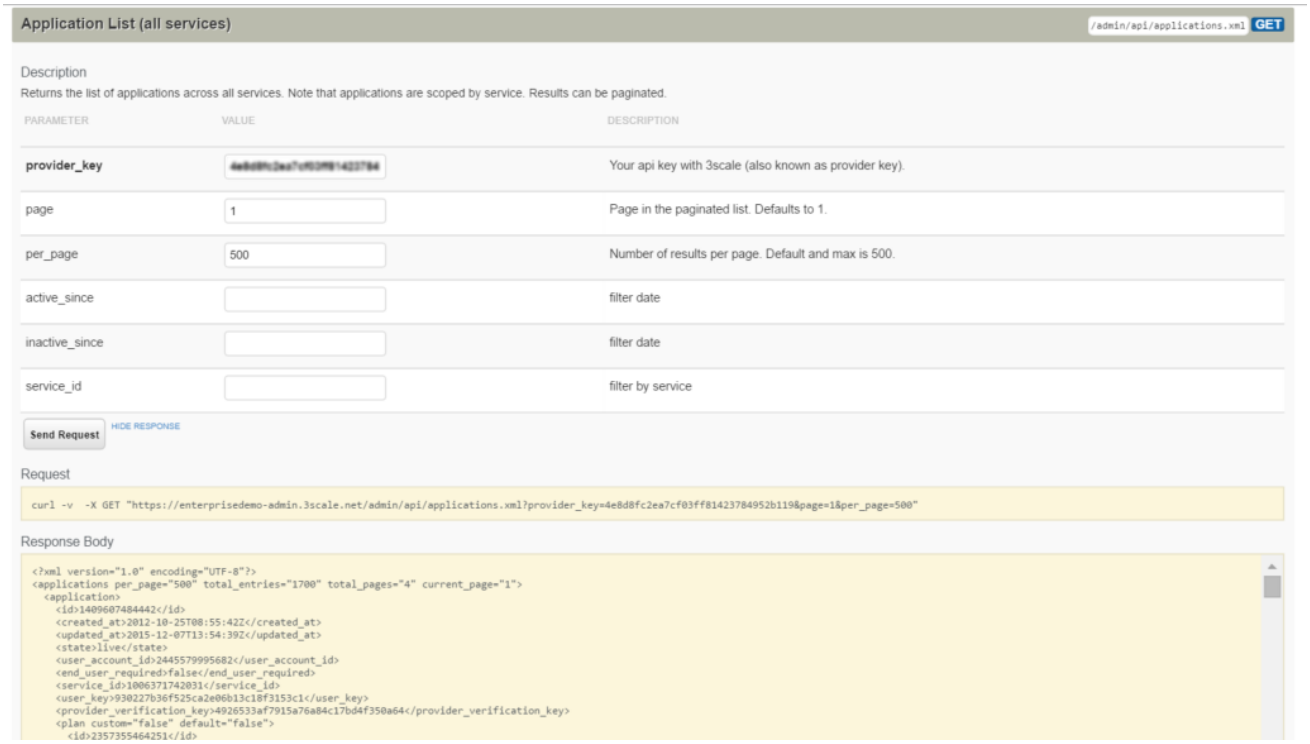
- ある期間 (10 日間のうち初めの 5 日間など) API を使用した後、使用を止めた開発者。これらの開発者は、使ってはみたがアクティブではなくなったグループです。
- API を継続的に使用している開発者。このグループについては、使用率の増加 (または減少) を知りたいと考えました。

この情報は、3scale の組み込み解析機能で利用可能です。問題は、集約データとして表示するビューがないことで、これは非常に面倒な状況です。

30.2. カスタムの手順による 3SCALE アプリケーションの解析値の抽出

アプリケーションの解析値を抽出するには、ActiveDocs の使用から作業を始めます。3scale ActiveDocs は、管理ポータルの **Account Settings > Integrate > 3scale API Docs**から利用可能です。3scale の API は、すべて ActiveDocs として利用可能です。したがって、ブラウザーから使用すること

ができます。これにより、ニーズに最適なリクエストを探し、リクエスト (curl に類似) を取得し、レスポンスを得ることができます。ActiveDocs の例を以下の図に示します。



これは、スクリプトが解析を抽出するすべてのアプリケーションを取得する API リクエストの ActiveDocs です。

- ActiveDocs の調査を完了したら、任意のスクリプト言語でリクエストを指定します。この例では Ruby を使用しています。
- 希望の処理を実行するスクリプトが書き上がるまで、この作業を繰り返します。解析機能の拡張を例に取ると、ここから スクリプトを確認することができます。これをご自分のアカウントで試すことができます。

ActiveDocs により、API のできることを素早く理解することができます。後は、実行したいタスクに必要な 3 つか 4 つのリクエストを探し、スクリプトを 1 つにまとめます。

以下の手順は、例のお客様が希望するカスタム解析機能を実現するステップを示しています。

手順

1. 全アプリケーションのリストを取得する。この操作にはページネーションが必要です。

```
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
   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. 条件を満たしていないアプリケーションを除外する (評価版のプランで、かつ 10 日前以降のものに絞り込む)。

```
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. 条件を満たすそれぞれのアプリケーションについて、その使用状況 (過去 10 日間のアプリケーションのヒットカウント) を取得する。

```
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. 開発者の情報はアカウントオブジェクトに保存されるため、アプリケーションをアカウントに紐付けする。

```
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. すべてを 1 つにまとめてスクリプトを完了する。このスクリプトで、3scale に組み込まれた解析機能ではまだ利用できなかった情報を取得することができます。完成したスクリプト を参照してください。

第31章 アプリケーションに対する 3SCALE の組み込みトラフィック解析の表示

サポートおよび拡張を保証する操作が必要なアカウントを特定するには、アプリケーションのトラフィックに関するビジュアル情報を表示します。API を使用する各アプリケーションには、3scale システムのトラフィックトレースがあり、管理ポータルから表示できるだけでなく、API によって復元することもできます。

前提条件

- 3scale の開発者アカウント
- 3scale アプリケーションおび API クレデンシャル

手順

1. 解析を表示するアプリケーションに移動します。
   Audience > Accounts > Listingまたは **Audience > Applications > Listing**の順に移動し、**Applications** タブでアプリケーションを特定することができます。あるいは、アプリケーションの特定 に記載のチュートリアルの手順に従って、アプリケーションを検索することができます。

2. アプリケーションを特定すると、以下の図に示すようなアプリケーションに関する情報が含まれる概要ページが表示されます。

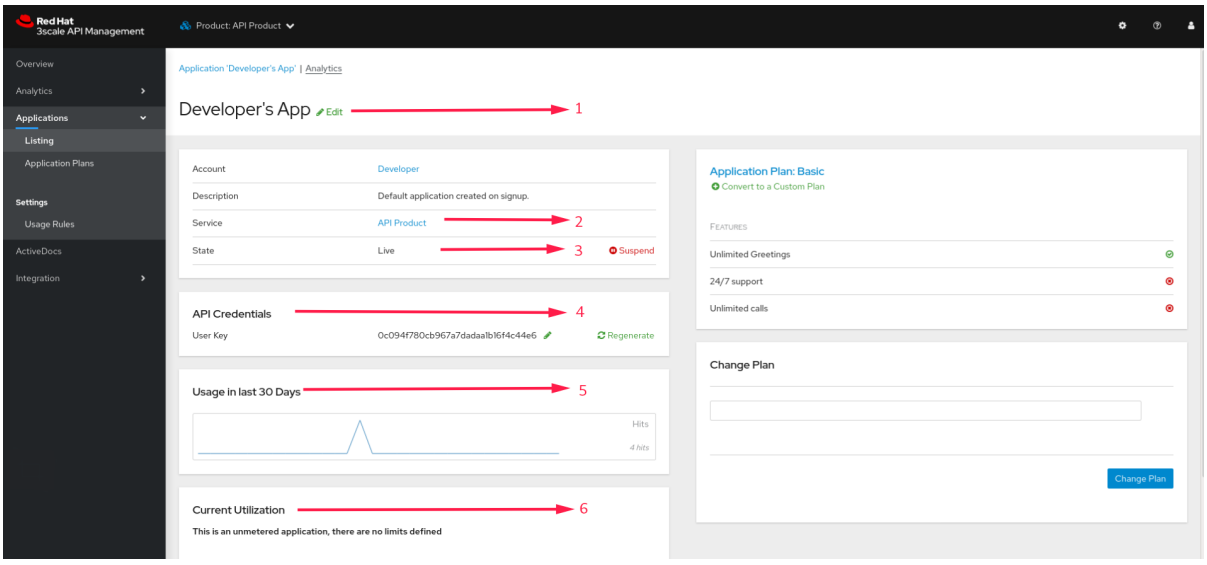


表31.1 図でラベルを付けた項目が、それぞれ以下の情報に対応しています。

番号	アプリケーション情報
1	開発者が指定したアプリケーションの名前
2	アプリケーション用に取得されたメタデータ (取得するデータの設定方法については、アドバンストセクションで説明します)
3	アプリケーションのステータス (Live または Suspended)

番号	アプリケーション情報
4	アプリケーションの現在有効な API 識別子、キー、および証明書。このビューは、API を 3scale に統合する際に使用する認証方法により異なります。
5	アプリケーションのトラフィックに関する統計の要約
6	アプリケーションに適用されるアプリケーションプランおよび適用される流量制御に関する情報

3. アプリケーション名の上にある **Analytics** のリンクをクリックします。アプリケーションの使用状況に関するチャートが表示されます。メトリクス、メソッド、および時間範囲を変更すると、アプリケーションに関するさまざまなタイプのデータを確認することができます。

第32章 API に対する 3SCALE レスポンスコードログの設定および評価

クライアントが API をどのように使用しているかを確認し、サーバーが想定どおりに実行されているかどうかをリアルタイムで確認するには、3scale のレスポンスコードログを設定して使用します。

手順

- Docker または OpenShift デプロイメントを使用する APIcast ゲートウェイを起動する際に、**APICAST_RESPONSE_CODES** 環境変数を **1** または **true** に設定します。これにより、レスポンスコードの追跡が可能になります。

例

有効にすると、APIcast ゲートウェイは、承認呼び出しに関してアップストリームサービスによって返される API レスポンスの HTTP ステータスコードを取得し、それらを Service Management API に送信します (**authrep** コール)。以下に例を示します。

```
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"
```

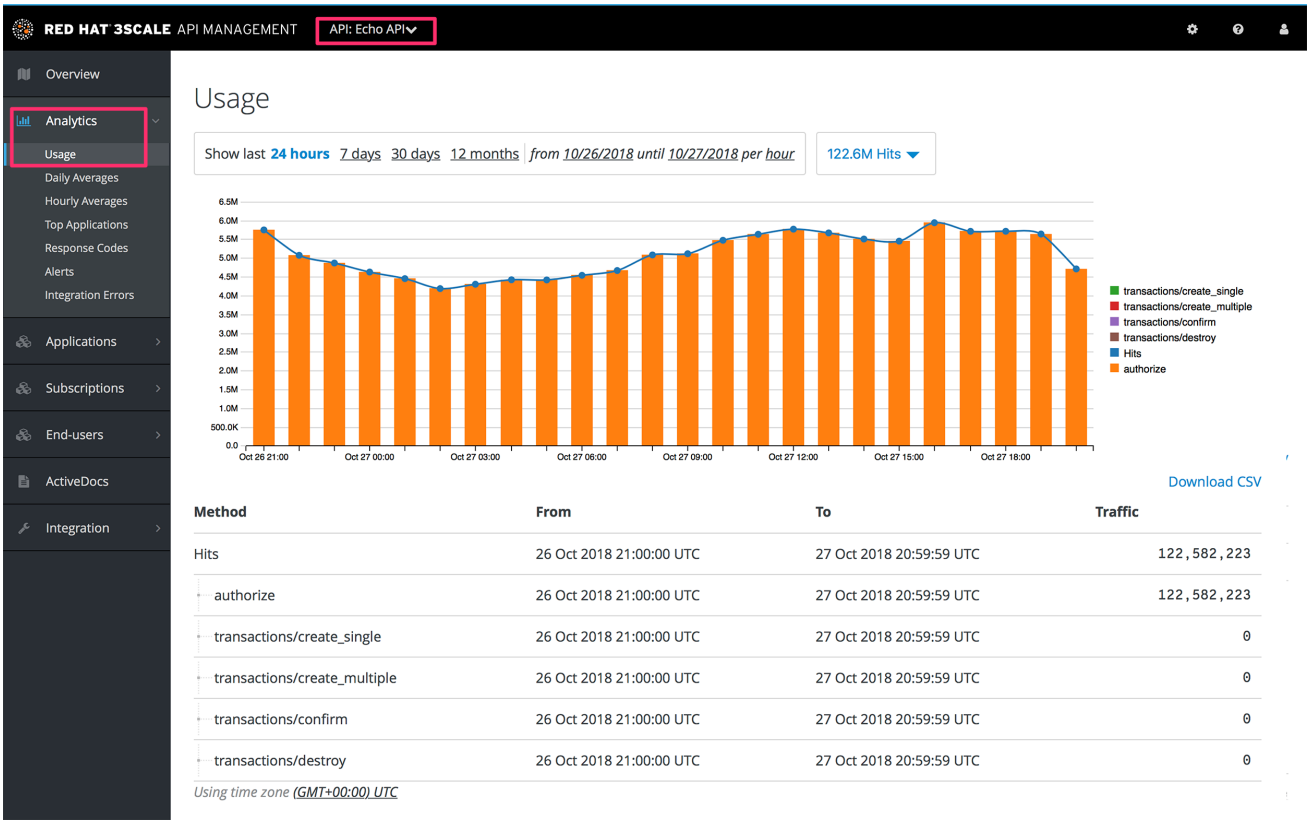
上記の例では、**log[code]=200** が送信されています (API が **200** のステータスコードと共に応答した)。

検証

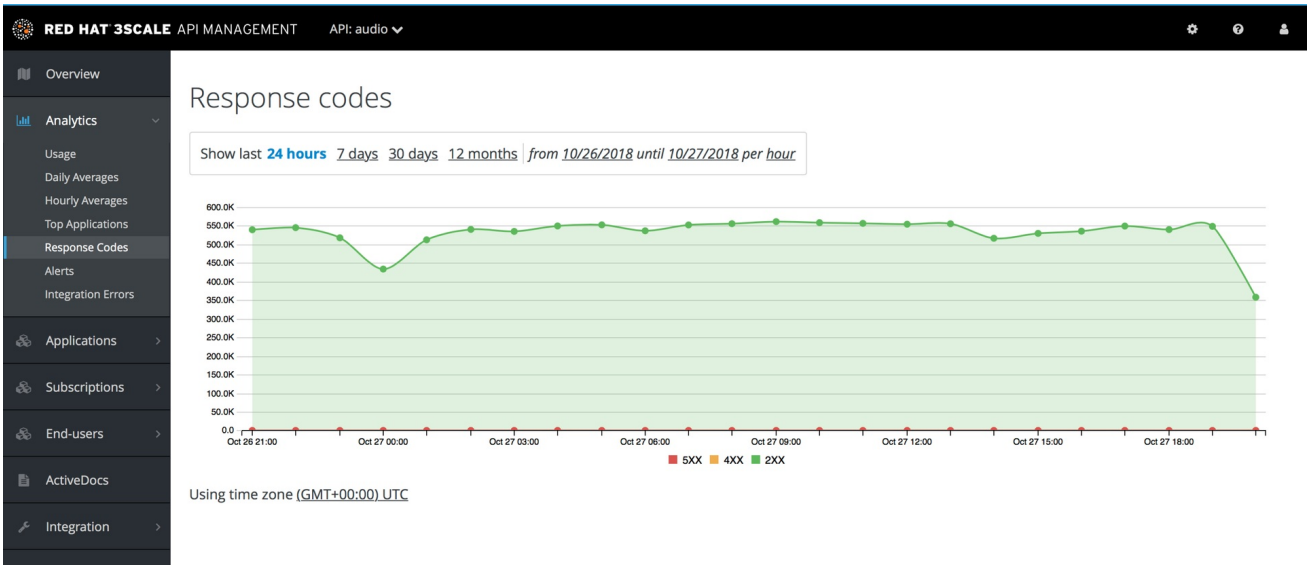
インテグレーションを確認するには、有効な 3scale クレデンシャルを使用して API プロダクトを呼び出してから、**Analytics > Usage** のページで呼び出しが正しく報告されたことを確認します。

注記

- 応答コード追跡は、すべての応答を正確にカウントすることを意図したものではありません。
- このビューの価値は、一定期間の傾向を視覚的に表現することです。
- レスポンスコードトラッキングおよび 3scale Auth Caching モード:**None** は、サポートされている組み合わせではありません。



ここまで問題がなければ、**Analytics >Response codes** のページに移動します。レスポンス (2xx、4xx、または 5xx) に応じて色分けされた最新トラフィックのグラフが表示されるはずです。



グラフツールにより、レスポンスコードの履歴を表示することができます。異なる期間および細かさのレベルでレスポンスコードの統計を確認することもできます。時間選択バーをクリックして、ニーズに合った期間と細かさを定義します。