



# Red Hat 3scale API Management 2.13

## デベロッパーポータル作成

適切なデベロッパーポータルを利用して API を確実に取り入れて簡単に作成する



## Red Hat 3scale API Management 2.13 デベロッパーポータルの作成

---

適切なデベロッパーポータルを利用して API を確実に取り入れて簡単に作成する

## 法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

本ガイドでは、Red Hat 3scale API Management 2.13 のデベロッパーポータルについて説明します。

## 目次

多様性を受け入れるオープンソースの強化 .....	4
<b>第1章 3SCALE 管理の API 用デベロッパーポータル作成の概要 .....</b>	<b>5</b>
1.1. デベロッパーポータル作成用の 3SCALE 編集環境の確認 .....	5
1.2. 3SCALE ネイティブデベロッパーポータルの変更 .....	7
1.3. ネイティブデベロッパーポータルを変更するための追加の 3SCALE 機能の説明 .....	10
1.4. 3SCALE のレイアウトとパーシャルを使用したデベロッパーポータルのコンテンツの再利用方法 .....	12
1.5. 3SCALE 管理の API 用デベロッパーポータルへのイメージおよびその他のアセットの追加 .....	12
1.6. 3SCALE 管理の API のネイティブデベロッパーポータルのカスタマイズに関する考慮事項 .....	13
1.7. 3SCALE 管理の API のデベロッパーポータルへのアクセスを許可するための要件 .....	14
1.8. 3SCALE 管理の API のデベロッパーポータルのカスタマイズに関するオプションステップ .....	16
1.9. 2.8 よりも前の 3SCALE リリースで使用されるデベロッパーポータルの更新 .....	17
<b>第2章 カスタムのサインアップフォームフィールド .....</b>	<b>20</b>
<b>第3章 サインアップフローの設定 .....</b>	<b>26</b>
3.1. すべての承認ステップの削除 .....	26
3.2. すべてのデフォルトプランの有効化 .....	27
3.3. ワークフローのテスト .....	28
<b>第4章 マルチサービスへのサインアップ .....</b>	<b>29</b>
4.1. 前提条件 .....	29
4.2. はじめに .....	29
4.3. マルチサービスへのサインアップ .....	29
<b>第5章 デベロッパーポータルの認証 .....</b>	<b>32</b>
5.1. ユーザー名/メールアドレスおよびパスワードによる認証の有効化と無効化 .....	33
5.2. GITHUB による認証の有効化および無効化 .....	33
5.3. AUTHO による認証の有効化および無効化 .....	35
5.4. RED HAT SINGLE SIGN-ON による認証の有効化および無効化 .....	35
<b>第6章 デベロッパーポータル用の RED HAT SINGLE SIGN ON .....</b>	<b>41</b>
6.1. 3SCALE プラットフォームでのユーザーの作成 .....	41
6.2. ログインリンクのリクエスト .....	41
6.3. 自動ログインによるユーザーのリダイレクト .....	41
<b>第7章 コンテンツの制限 .....</b>	<b>43</b>
7.1. ページの制限 .....	43
7.2. コンテンツブロックの制限 .....	44
7.3. 追加フィールドの設定の自動化 .....	44
7.4. ユーザーログインの要求 .....	45
<b>第8章 メールテンプレート .....</b>	<b>46</b>
8.1. メールテンプレートのカスタマイズ .....	46
8.2. 補足情報 .....	47
<b>第9章 LIQUID: デベロッパーポータル .....</b>	<b>48</b>
9.1. デベロッパーポータルでの LIQUID の使用 .....	48
9.2. メールテンプレートでの LIQUID の使用 .....	49
9.3. トラブルシューティング .....	49
<b>第10章 LIQUID: メールテンプレート .....</b>	<b>51</b>
10.1. アカウント管理 .....	51
10.2. クレジットカードに関する通知 .....	52

---

10.3. 制限に対するアラート	52
10.4. アプリケーション	52
10.5. 請求	53
10.6. サービス	54
10.7. サインアップ	55
<b>第11章 デベロッパーポータルのレイアウトのカスタマイズ</b> .....	<b>56</b>
11.1. 新規 CSS ファイルの作成	56
11.2. ページレイアウトへのスタイルシートのリンク	56
11.3. ページレイアウトテンプレートの定義	56
<b>第12章 組み込みページの変更</b> .....	<b>57</b>
12.1. 要素の特定	57
12.2. 要素の変更または非表示	58
12.3. オプション A: CSS	58
12.4. オプション B: JQUERY	58
<b>第13章 契約条件の設定</b> .....	<b>60</b>
13.1. 契約条件	60
13.2. クレジットカードポリシー	61



## 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。



## 第1章 3SCALE 管理の API 用デベロッパーポータル作成の概要

3scale デベロッパーポータルは、API 利用者が使用する Web サイトです。

- お客様の 3scale 管理のアップストリーム API にアクセスするためにサインアップする
- お客様のアップストリーム API の使用方法に関するドキュメントを参照する

3scale が提供するデベロッパーポータルのサンプルには、ほとんどの API プロバイダーがデベロッパーポータルに実装する必要のある機能が含まれます。このネイティブデベロッパーポータルでは、サンプルの Echo API を使用して、一般的なデベロッパーポータルの構造を実証します。ネイティブデベロッパーポータルの確認後、ポータルを変更するための手順が記載されているので、独自のデベロッパーポータルの作成方法を理解することができます。

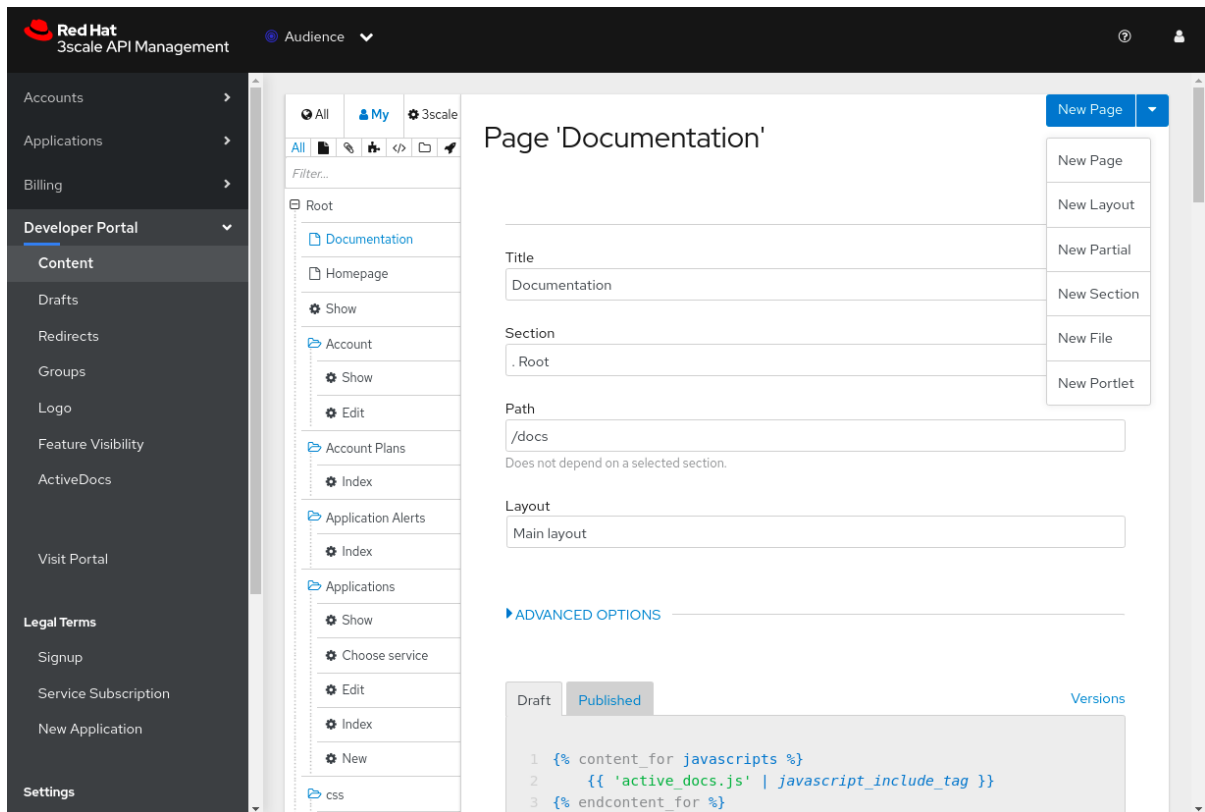
ネイティブデベロッパーポータルの確認および変更の前提条件はありません。ただし、ネイティブのデベロッパーポータルを変更して独自のデベロッパーポータルを作成したら、デベロッパーポータルを API 利用者に公開する前に、サインインワークフローおよび認証も実装する必要があります。

### 1.1. デベロッパーポータル作成用の 3SCALE 編集環境の確認

デベロッパーポータルの作成を開始する前に、3scale で提供されるサンプルの Echo API デベロッパーポータルを確認してください。Echo API デベロッパーポータルは、専用のデベロッパーポータルを作成するための開始点となります。デベロッパーポータルはゼロから作成しません。代わりに、ネイティブの Echo API デベロッパーポータルを変更して、希望のルックアンドフィールのデベロッパーポータルを作成します。

#### 手順

1. 3scale 管理ポータルで、上部のコンテキストセクターを展開し、**Audience** をクリックします。
2. 左側のナビゲーションツリーで **Developer Portal** を展開し、**Content** をクリックします。これにより、デベロッパーポータルを作成するためのメインの編集環境が表示されます。



3scale では、**Root** に、デベロッパーポータルのリソース階層が表示されます。

- a. **Documentation**、**Homepage**、および **Show** は、デベロッパーポータルの基盤ページです。各ページでスクロールダウンし、ページコンテンツを定義する HTML を表示します。
  - b. これらのページより下のフォルダーには、3scale リソースをデベロッパーポータルにプルするページが含まれます。たとえば、**Account** フォルダーには、3scale 管理者が管理ポータルで作成した 3scale アカウントを表示および編集するページが含まれます。これらのページを開始点として使用し、必要に応じて変更します。
  - c. 右上の **New Page** ドロップダウンでは、ページ、レイアウト、パーシャル、セクション、ファイル、またはポートレットを追加できます。それぞれの項目を選択して、作成するために指定する情報を表示します。
3. **Developer Portal > Content** を選択し、**Layouts** が表示されるまでリソースの階層をほぼ最下部までスクロールし、**Main layout** をクリックします。
  4. **Layout 'Main Layout'** の見出しが表示されるように、上方向にスクロールします。内部タイトルとシステム名の後に、**Liquid** が有効になっていることを確認できます。Liquid は、3scale が 3scale システムからのほとんどのデータを表示および処理するのに使用するフレームワークです。ページコンテンツを定義するコードには、Liquid マークアップと HTML が含まれます。これは **Draft** タブに表示されます。これには、デベロッパーポータルページのメインレイアウトのコードが含まれています。
  5. 左側のナビゲーションツリーで、**Developer Portal > Content** の下にある各サブカテゴリーを順番にクリックして確認します (**Drafts**、**Redirects**、**Groups**、**Logo**、**Feature Visibility**、および **ActiveDocs**)。
  6. 左側のナビゲーションツリーで、**Developer Portal** の下にある **Visit Portal** (最後のエントリー) をクリックします。  
新しいブラウザタブに、3scale が提供する Echo API デベロッパーポータルの開発バージョンの Web サイトが表示されます。この開発バージョンを使用して、3scale ネイティブデベロッパーポータルを表示します。その後、ネイティブデベロッパーポータルのカスタマイズを繰り返

返し、加えた変更を表示して、独自のデベロッパーポータルを作成することができます。

最上部に**Draft|Published**と書かれた濃いグレーのパネルが右側にあるので、これがデベロッパーポータルの開発バージョンであることが確認できます。Draft ビューは、反復/差分の改善をサポートします。Draft バージョンの見栄えおよび動作が希望どおりになったら、公開できます。

右側のパネルには、現在のページのコンテンツを提供する要素が一覧表示されます。

- a. Page Homepage
- b. Layout Main layout
- c. Partial Submenu
- d. Partial analytics

7. **Page Homepage** をクリックします。新しいブラウザタブに、編集するために **Page 'Homepage'** が開いているデベロッパーポータルの編集環境が表示されます。

8. Echo API デベロッパーポータルの開発バージョンに戻り、右上の **SIGN IN** をクリックすると、API 利用者がデベロッパーポータルにサインインするために使用する **SIGN IN** ページが表示されます。

右側のグレーのパネルのテンプレートリストの下に、ユーザー名およびパスワードがあります。これを使用して、デベロッパーポータルへのサインインをシミュレーションできます。

- a. **SIGN IN** ページの **USERNAME OR EMAIL** フィールドに **John** を入力します。これは、右側のグレーパネルに一覧表示されるユーザー名です。
- b. **PASSWORD** フィールドに、グレーのパネルに一覧表示されるパスワードである **123456** を入力します。
- c. **Sign In** をクリックし、API 利用者に表示されるデベロッパーポータルを表示します。

## 次のステップ

引き続き、十分にネイティブデベロッパーポータルを確認してください。編集環境や、Echo API デベロッパーポータルの開発バージョンの操作に慣れたら、[3scale ネイティブデベロッパーポータルの変更](#)の手順に従います。

## 1.2. 3SCALE ネイティブデベロッパーポータルの変更

Echo API デベロッパーポータルを確認したら、独自のデベロッパーポータルの作成を開始する前にいくつかの変更を加えます。これらの練習のステップは、デベロッパーポータル作成の準備に役立ちます。

この手順では、サンプルの 3scale Echo API ランディングページ見出しを、一般的な Swagger Petstore API の見出しに置き換えます。また、デベロッパーポータルを更新して Petstore API のドキュメントを表示する方法についても説明します。

### 手順

1. 3scale 管理ポータルで、上部のコンテキストセクターを展開し、**Audience** をクリックします。
2. 左側のナビゲーションツリーで **Developer Portal** を展開し、**Content** をクリックします。

3. **Root** で、デベロッパーポータルのランディングページの内部タイトルである **Homepage** をクリックし、デベロッパーポータルの開発バージョンに表示されるランディングページの見出しを変更します。
  - a. **Page 'Homepage'** で、ランディングページをレンダリングするコードが表示されるまでスクロールダウンします。
  - b. **5** 行目で、

```
<h1>Echo API</h1>
```

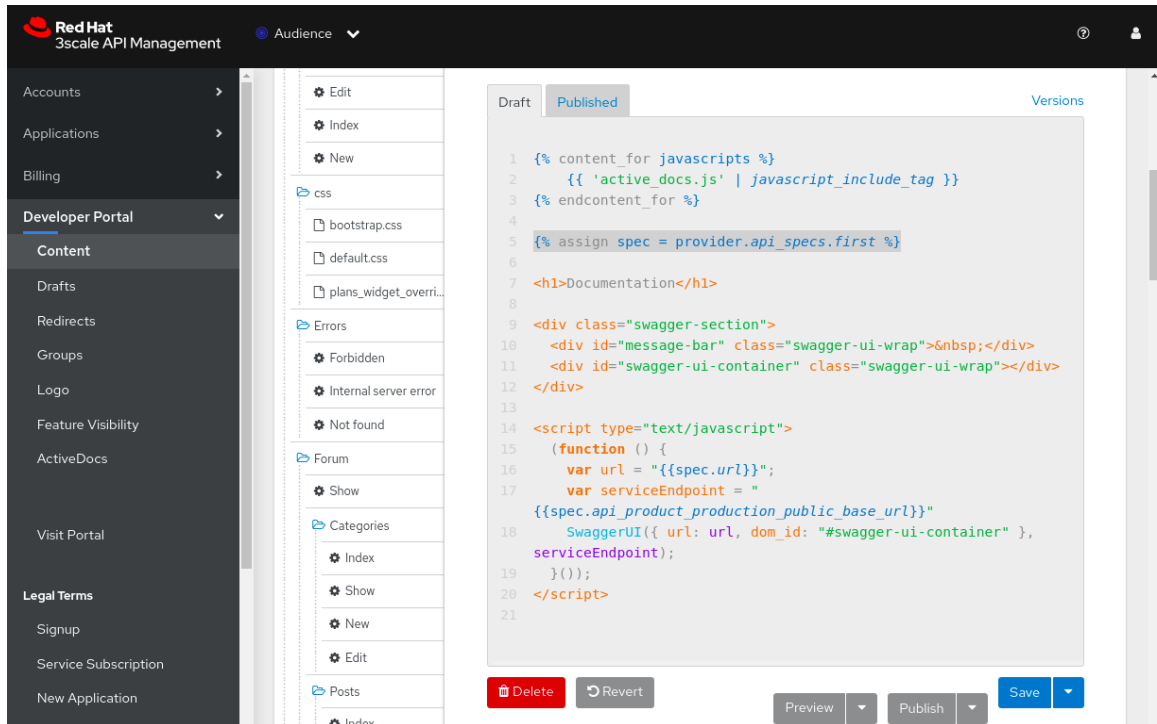
上記を以下のように変更します。

```
<h1>Petstore API</h1>
```
  - c. ページの下部で、**Publish** をクリックします。
  - d. 左側のナビゲーションツリーの **Developer Portal** の下で、**Visit Portal** をクリックしてデベロッパーポータルの開発バージョンを表示し、現在ランディングページ見出しが **Petstore API** であることを確認します。
4. デベロッパーポータルの開発バージョンのまま、トップメニューバーの **Documentation** をクリックします。デベロッパーポータルに、Echo API の ActiveDocs が表示されます。
5. 3scale 管理ポータルに戻り、**Developer Portal > ActiveDocs** を選択して Echo API のエントリーを表示します。3scale では、Echo API を定義する OpenAPI ドキュメントが提供されません。3scale はこの OpenAPI ドキュメントを使用して Echo API の ActiveDocs を表示します。
6. Swagger Petstore API を定義する OpenAPI ドキュメントをインポートします。
  - a. <https://petstore.swagger.io/v2/swagger.json> にアクセスし、JSON コンテンツをクリップボードにコピーします。
  - b. 3scale 管理ポータルに戻り、**Developer Portal > ActiveDocs** を選択します。
  - c. **ActiveDocs** ページで、**Create a new spec** をクリックします。
  - d. **Name** フィールドに **Petstore** を入力します。
  - e. **Publish?** を選択します。
  - f. **API JSON Spec** ウィンドウをクリックし、クリップボードにコピーから Swagger Petstore JSON コンテンツを貼り付けます。
  - g. ページの下部にある **Create Spec** をクリックします。3scale に、Petstore API の ActiveDocs が表示されます。
  - h. 左側のナビゲーションツリーで、**Developer Portal** の下にある **ActiveDocs** をクリックします。Echo API の後に、**Petstore** の 2 番目のエントリーがあります。
7. デベロッパーポータルに Petstore API のドキュメントを表示します。
  - a. 左側のナビゲーションツリーで、**Developer Portal** の下にある **Content** をクリックします。
  - b. **Root** で、デベロッパーポータルのドキュメントページの内部タイトルである **Documentation** をクリックします。

- c. Page 'Documentation' で、ドキュメントのランディングページをレンダリングするコードが表示されるまでスクロールダウンします。5 行目は、デベロッパーポータルが ActiveDocs を表示する OpenAPI ドキュメントを識別します。5 行目のデフォルト値は、以下のとおりです。

```
{% assign spec = provider.api_specs.first %}
```

デフォルトの動作では、デベロッパーポータルは Developer Portal > ActiveDocs ページの最初のエントリーの ActiveDocs を表示します (初期は Echo API)。以下の図では、5 行目がハイライトされています。



- d. 5 行目を変更し、`provider.api_specs.first` を変更し、ActiveDocs ページの 2 番目のエントリーを識別するインデックスを追加します。

```
{% assign spec = provider.api_specs[1] %}
```

デフォルトの動作では、デベロッパーポータルは 1 つの OpenAPI ドキュメントの ActiveDocs のみを表示します。複数の OpenAPI ドキュメントの ActiveDocs を表示するには、この簡単な変更以外に **Documentation** ページを変更する必要があります。

- e. ページの下部で、**Publish** をクリックします。
8. 左側のナビゲーションツリーの **Developer Portal** の下で、**Visit Portal** をクリックしてデベロッパーポータルの開発バージョンを表示します。
  9. トップメニューバーで **Documentation** をクリックし、Swagger Petstore のドキュメントを参照してください。

## 次のステップ

デベロッパーポータルの作成を開始するには、3scale 管理の API へのアクセス情報および API のドキュメントを表示するようにネイティブのデベロッパーポータルページを変更します。

### 1.3. ネイティブデベロッパーポータルを変更するための追加の 3SCALE 機能の説明

3scale は、専用のデベロッパーポータルを作成するための、ネイティブのデベロッパーポータルをカスタマイズする多くの機能を提供します。

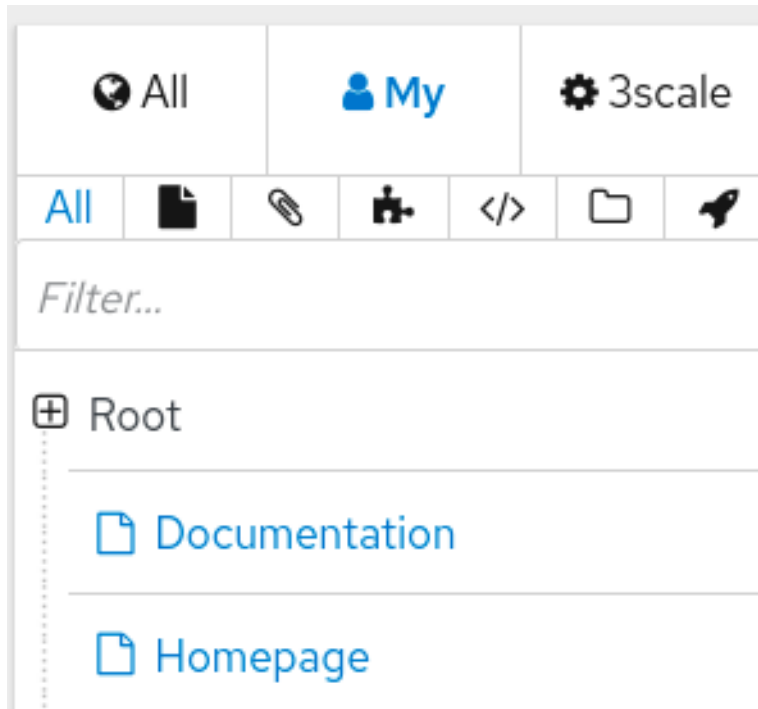
デベロッパーポータルの開発中、デベロッパーポータルを表示する必要のあるメンバーは、アクセスコードを指定する必要があります。API プロバイダーが API 利用者にデベロッパーポータルへのアクセスを公開するために必要なタスクを実行している間、このアクセスコードを使用し続けます。このアクセスコードを使用することで、ご自分やこのコードを持つメンバーだけがデベロッパーポータルを表示できるようになります。アクセスコードは **Domains & Access** ページにあります。このページは、管理ポータルで **Audience > Developer Portal > Domains & Access** の順に選択して表示できます。

以下の一覧で、ネイティブデベロッパーポータルを変更するための機能の一部を紹介します。

- **Developer Portal > Content** 環境では、右上の **New Page** ドロップダウンにより、デベロッパーポータルに以下の要素を追加できます。
  - **Page** - デベロッパーポータルのコンテンツの基本単位。
  - **layout**: 複数のページが使用できるテンプレート。
  - **partial**: 他のパーシャル、レイアウト、およびページ内で再利用できるコンテンツ。
  - **Section**: ディレクトリーと機能的に同等です。デベロッパーポータルのコンテンツを整理するためのセクションを作成します。
  - **File**: スタイルシート、イメージ、スクリプトなど、デベロッパーポータルで使用するリソース。
  - **Portlet**: 外部 RSS フィード、目次、または最新のフォーラム投稿。
- HTML、Markdown、または Textile でコードを入力できます。コードエディターは、コードの強調表示、表作成、行番号、その他の機能をサポートします。
- ページのドラフトまたは公開バージョンをプレビューできます。ページのテキストエディターの下で、**Preview** をクリックするとデベロッパーポータルの開発バージョンが表示されます。プレビューには、右側に濃いグレーの垂直バーが表示されます。濃いグレーのバーの上部で **Draft | Published** のいずれかが強調表示され、表示しているバージョンが示されます。このバーには、以下の項目の編集環境へのリンクが含まれます。
  - ページ。
  - ページが使用するレイアウト。
  - ページが使用するパーシャル。パーシャルとは、異なるページの多くの場所で再利用できるコードのことです。

デベロッパーポータルの開発バージョンの Draft ビューは、イテレーティブ/差分の改善をサポートします。Draft バージョンの見栄えおよび動作が正しかったら、公開できます。アクセスコードを使用すると、ページを公開すると、デベロッパーポータルが公開された場合に API 利用者にどのように見えるかを確認できます。アクセスコードが使用されない場合、ページを公開すると、ライブのデベロッパーポータルでそのページが更新されます。

- フィルターフィールドを使用すると、デベロッパーポータル環境のリソースを検索できます。3scale は、検索対象の要素のみを表示するので、作業対象を簡単に見つけることができます。フィルターフィールドは、アイコンのすぐ下にあります。



- **Developer Portal > Redirects** ページでは、あるデベロッパーポータル URL から別のデベロッパーポータル URL へのリダイレクトを設定できます。たとえば、作成したページを非推奨にした場合は、リクエストを新しいページにリダイレクトすることができます。
- プレースホルダーは編集可能なフィールドです。たとえば、**開発者ポータル > コンテンツ > ログイン > 新規** ページには、いくつかのプレースホルダーがあります。

```
<input id="session_username" name="username" tabindex="1"
  autofocus="autofocus" type="text"
  placeholder="Authenticate with your email"
  class="form-control">
...
<input id="session_password" name="password" tabindex="2"
  type="password"
  placeholder="...and password"
  class="form-control">
```

このコードは、開発者ポータルで次のフィールドを生成します。

USERNAME OR EMAIL	<i>Authenticate with your email</i>
PASSWORD	<i>...and password</i>

プレースホルダーテキストを更新して公開し、開発者ポータルで更新を確認できます。たとえば、**Authenticate with your email** を **Authenticate email** に変更できます。ページを公開すると、開発者ポータルで更新されたプロンプトを確認できます。次に例を示します。



USERNAME OR EMAIL

Authenticate email

PASSWORD

...and password

## 1.4. 3SCALE のレイアウトとパーシャルを使用したデベロッパーポータルのコンテンツの再利用方法

Developer Portal > Content環境で、リソース階層をスクロールダウンすると、Layouts と Partial の見出しが表示されます。レイアウトとパーシャルを使用すると、デベロッパーポータルのコンテンツを再利用できます。

- レイアウトは、ページの基本構造を定義し、ページのテンプレートとして機能します。特定のレイアウトを使用するすべてのページには、レイアウトが定義するコンテンツが含まれます。Layouts のリソースの階層で分かるように、ネイティブデベロッパーポータルでは Error layout と Main layout が提供されます。これらのレイアウトのコードを変更し、レイアウトを作成できます。
- パーシャルは、デベロッパーポータルの複数の場所で使用するコードを定義します。たとえば、パーシャルは、全レイアウトのフッターや、複数のページのサイドバーを定義できます。ページ、レイアウト、別のパーシャル、またはメールテンプレートでパーシャルを使用できます。リソース階層の Partial で、ネイティブデベロッパーポータルが提供するパーシャルを確認できます。解析値の収集、アプリケーションプランのサブスクライブ、サブメニューの表示用のパーシャルがあります。また、これらのパーシャルを変更したり、パーシャルを作成したりすることができます。  
ページ、レイアウト、パーシャル、またはメールテンプレートでパーシャルを使用するには、パーシャルの名前と共に **include** コマンドを指定します。たとえば、**submenu** パーシャルを使用するには、以下のように指定します。

```
{% include "submenu" %}
```

レイアウトとパーシャルには、ドラフト状態と公開状態、およびバージョン履歴があります。たとえば、ドラフトを保存し、公開し、最後に公開したバージョンに戻すことができます。

レイアウトとパーシャル用のコードの編集および書き込みについては、[3scale Liquid リファレンス](#) を参照してください。

## 1.5. 3SCALE 管理の API 用デベロッパーポータルへのイメージおよびその他のアセットの追加

通常、独自のイメージ、ファイル、またはその他のアセットを使用して、ネイティブデベロッパーポータルをカスタマイズします。そのためには、デベロッパーポータルのコンテンツライブラリーにファイルを追加し、この環境でパスをメモしてから、コードでコンテンツライブラリーのファイルの場所へのリンクを追加します。

### 前提条件

- デベロッパーポータルで使用するイメージ、ファイル、またはその他のアセットへのアクセス

### 手順



1. アセットをデベロッパーポータルのコンテンツライブラリーに追加します。
  - a. **Developer Portal > Content**環境で、右上の **New Page** を展開し、**New File** を選択します。
  - b. **Upload File** で **Section** フィールドをクリックし、追加するアセットを保存するディレクトリの一覧を表示し、適切なディレクトリをクリックして選択します。たとえば、**images** を選択します。
  - c. オプション:**Path** フィールドに、役に立つ追加のパスレベルを入力します。
  - d. **Downloadable** を選択します。
  - e. **Choose File** をクリックして、追加するファイルに移動し、それを選択して **Open** をクリックします。
  - f. オプション:**Tag list** フィールドに、この新しいコンテンツをコードに含めるために使用する Liquid タグを入力します。たとえば、**my-logo** を入力します。
  - g. **Create File** をクリックして、デベロッパーポータルのコンテンツライブラリーにアセットを追加します。
2. 追加したアセットへのパスを書き留めます。たとえば、**MyLogo.png** イメージを **images** セクションに追加した場合、パスは **/images/MyLogo.png** になります。
3. コードでアセットを使用するには、HTML **<a>** 要素を挿入します。以下に例を示します。

```
<a href="/images/MyLogo.png"/>
```

または、Liquid タグをアセットに追加した場合は、以下のようにタグを指定してコードでアセットを使用できます。

```
{% my-logo %}
```

## 1.6. 3SCALE 管理の API のネイティブデベロッパーポータルのカスタマイズに関する考慮事項

ネイティブデベロッパーポータルをカスタマイズして独自のデベロッパーポータルを作成する前に、基本要素、再利用可能なコンテンツ、ページ階層、ページヘッダー/フッター、およびブランディングを計画する必要があります。

- 基本的なデベロッパーポータル要素: デベロッパーポータルのプランには、以下を含める必要があります。
  - サイトマップ: ポータルがどのように設定されているかのスケルトン。
  - 上部のメニューバー: 各ページで繰り返されるナビゲーション。
  - サイドメニューバー: 各セクションの個々のページへのアクセス。
  - ページレイアウトのガイドライン: 一貫したルックアンドフィール用。
  - 再利用可能なコンテンツ: レイアウトとパーシャルにより、デベロッパーポータルの一貫性が確保されます。ページの作成を開始する前に、作成します。

- ページ階層: サイトマップのルートレベルで開始し、トップメニュー項目ごとにセクションを作成します。  
次に、各トップメニュー項目に必要なページを作成します。ページを作成する際には、必ず正しいセクションに関連付けてください。これにより、デベロッパーポータルの URL パスに対して一貫性のある設定が作成されます。Textile や Markdown などのマークアップ言語でページをコーディングするには、ページの **ADVANCED OPTIONS** を展開します。Handler フィールドはマークアップ言語を識別します。
- ページヘッダーとフッター: 通常、繰り返し表示されるページ要素はパーシャルで定義されます。レイアウトが少しだけの場合、パーシャルではなくレイアウトで直接ヘッダーとフッターを定義する方が望ましい場合があります。
- ブランディング: デフォルトのネイティブデベロッパーポータルスタイルシート **default.css** は大きく複雑です。このスタイルシートを拡張するのではなく、デフォルトを上書きするカスタム設定の専用のスタイルシートを作成します。  
ページの作成と同じ方法で、新しいスタイルシートを作成することができます。高度なページ設定で **css** セクションと適切な MIME コンテンツタイプを選択します。レイアウトで **default.css** へのリンクの後に、カスタムスタイルシートへのリンクを追加します。たとえば、**my-custom.css** を作成する場合は、以下を挿入します。

```
<link rel="stylesheet" href="/css/my-custom.css" />
```

- 変更できない組み込み機能には、次のものがあります。
  - ページネーション
  - メッセージメニュー
  - フォーラム - レイアウトのみを変更できます。
  - お支払い - これらのページでは、いくつかのテキストフィールドのみが編集できます。

## 1.7. 3SCALE 管理の API のデベロッパーポータルへのアクセスを許可するための要件

API 利用者にデベロッパーポータルへのアクセス権限を付与する前に、API プロバイダーは以下のタスクを実行する必要があります。これらのタスクは同時に実行できます。

- 必要な外観と操作感になるように、3scale ネイティブデベロッパーポータルを変更します。
  - [Liquids: デベロッパーポータル](#) では、Liquid マークアップを使用して API に関連する 3scale システムデータを表示し、処理する方法を説明しています。Liquid マークアップは、デベロッパーポータルページにロジックを追加する主要な方法です。
  - [デベロッパーポータルのレイアウトのカスタマイズ](#) では、独自のブランディングに一致するようにネイティブデベロッパーポータルを変更する方法について説明します。カスタマイズを簡単に行うための出発点として、標準のカスケードスタイルシート (CSS) を使用できます。
  - [組み込みページの変更](#) では、CSS および JavaScript を使用してシステムが生成するページの要素を変更または非表示にする方法を説明します。ネイティブ開発者ポータルの一部として提供されるすべてのページは、システム生成ページと呼ばれます。

HTML、CSS、Liquid、および Web サイト全般に精通している開発者は、ネイティブ Echo API デベロッパーポータルを変更してデベロッパーポータルの Web ページを作成することができます。この開発者は、ページを作成して、システムが生成したページのコードを変更して、API

利用者がデベロッパーポータルで表示するものを自由に作成することができます。

- 3scale API プロダクト、バックエンド、アプリケーションプランを定義し、ポリシーをプロダクトに割り当てます。  
3scale プロダクトは、API を利用者に公開する 3scale リソースです。デベロッパーポータルでは、API 利用者はプロダクトのドキュメントを読み、プロダクトが提供する API の使用にサブスクライブします。3scale プロダクトには、以下の特徴があります。
  - 1つまたは複数のバックエンド (作成した内部 3scale API) をバンドルします。
  - 制限、課金、および利用可能な機能に関してプロダクトを使用するためのルールを定義するアプリケーションプランがあります。
  - API 利用者の呼び出しを API に送信する前にゲートウェイが処理する方法に関する APIcast ゲートウェイ設定があります。プロダクトに追加するポリシーにより、デフォルトの APIcast ゲートウェイ動作が変更されます。

[管理ポータルガイド](#) を参照してください。

- 3scale 管理の API を定義し、文書化する OpenAPI ドキュメントをインポートします。  
デベロッパーポータルの基盤は、API を定義する OpenAPI ドキュメントです。OpenAPI ドキュメントを 3scale にインポートすると、3scale は ActiveDocs を作成または更新するため、API に関する有効なドキュメントをすぐに用意することができます。デベロッパーポータルで、API 利用者はこのドキュメントを使用して API を調べ、テストし、統合します。

OpenAPI ドキュメントに定義されている各オペレーションについて、ドキュメントをインポートすると、3scale はメソッドとマッピングルールを作成します。メソッドおよびマッピングルールは、API 利用者の API へのアクセスに対する制限およびルールを適用するのに役立ちます。

3scale ドキュメント [デベロッパーポータルでの API の提供](#) には、3scale および OpenAPI ドキュメントを操作するための情報および手順が記載されています。特に、[3scale OpenAPI 仕様として使用する OpenAPI ドキュメントの作成方法](#) および [ActiveDocs の 3scale への追加](#) を参照してください。

- API 利用者が 3scale 管理の API にアクセスするためにサインアップする方法のワークフローを設定します。  
サインアップのワークフローは、デベロッパーポータルでの API 利用者体験の最も重要な部分です。ワークフローは、セルフサービスから、誰が何にアクセスできるかを完全に制御するものまで、多岐にわたります。アカウント、サービス、およびアプリケーションプランにより、複数のレベルの粒度を提供します。それぞれのレベルで、管理する承認ゲートがあるかどうかと、API 利用者が何らかの選択を要求されるかどうかを制御できます。

最大限の自動化とセルフサービスに対応するために、すべての承認ステップを削除し可能なデフォルトプランをすべて有効にすることができます。サインアップ後すぐに、API 利用者のデベロッパーポータルへのアクセスを提供するキーを発行できます。

ネイティブデベロッパーポータルは、ユーザー、アカウント、アプリケーションのサインアップに関して一般的に使用されるフィールドを提供します。一般的に使用されるこれらのフィールドに、カスタムフィールドを追加する必要がある場合があります。詳細は、[サインアップフローの設定](#) および [カスタムのサインアップフォームフィールド](#) を参照してください。メールテンプレートをカスタマイズする前に、必ずワークフローを設定してください。

- API 利用者の認証を実装します。  
API 利用者のデベロッパーポータルへのアクセスを認証することにより、デベロッパーポータルのリソースおよび API が保護されます。以下の方法のいずれかを使用して、デベロッパーポータルへのアクセスを認証することができます。

- ユーザー名、電子メール、およびパスワード
- GitHub
- Auth0
- Red Hat Single Sign-On (RH-SSO)

[デベロッパーポータルの認証](#) を参照してください。

- デベロッパーポータルと API 利用者間のメール通信用に、3scale ネイティブテンプレートをカスタマイズします。

サインアップ時のアカウントアクティベーションリンクの提供、パスワードの復元、サービス料金、変更通知など、多くのさまざまなイベントで、デベロッパーポータルと API 利用者間の通信が必要です。3scale では、デベロッパーポータルが API 利用者へ送信する標準的な電子メールタイプごとにテンプレートが提供されています。

サインアップワークフローを定義したら、電子メールメッセージの内容をカスタマイズします。これにより、デベロッパーポータルで設定したワークフローに密接に一致させることができます。

[電子メールテンプレート](#) および [Liquid: 電子メールテンプレート](#) を参照してください。

- API 利用者が 3scale 管理の API にアクセスするために合意しなければならない規定、条件、およびポリシーを指定します。  
API 利用者がサインアップして API を呼び出すのを許可する場合、通常はアクセス権限を付与する前に規定、条件、およびポリシーに同意してもらいます。登録するため、特定のアプリケーションを使用するため、または特定のサービスを使用するため (デベロッパーポータルが複数のサービスを提供する場合) など、さまざまなバージョンの契約条件を設定することができます。

API の使用を有料にする場合は、おそらくクレジットカードポリシーへの同意も必要になります。

[契約条件の設定](#) を参照してください。

- API 利用者の請求およびクレジットカードゲートウェイを設定します。  
3scale の請求プロセスは毎日実施されます。有料サービスにサブスクライブしている API 利用者アカウントごとに請求書を作成します。請求書のステータスは、open、finalized、pending、unpaid、paid、failed、cancelled のいずれかです。3scale では、請求書の処理用に設定する支払いゲートウェイを使用します。

3scale の請求プロセスは、前払いモードまたは後払いモードで実施することができます。3scale における請求は暦月に基づき、月の初日に特別なイベントが発生します。

[管理ポータルガイドの請求](#) を参照してください。

デベロッパーポータルを実稼働環境に移行する前に実行する必要がある最後のタスクは、アクセスコードを削除することです。これは、デベロッパーポータルへのアクセスの認証を設定し、デベロッパーポータルを細部にわたってテストして、希望どおりに動作していることを確認した後にのみ行います。

アクセスコードを削除するには、**Developer Portal > Content**環境を表示します。右下の **Open your Portal to the world** をクリックして、この操作を確認します。

## 1.8. 3SCALE 管理の API のデベロッパーポータルのカスタマイズに関するオプションステップ

デベロッパーポータルを API 利用者に公開する要件に加えて、以下が必要になる場合があります。

- デベロッパーポータルで複数の API を提供する。  
3scale にインポートする各 OpenAPI ドキュメントは、個別の API オファリング (個別のサービス) の基盤を提供します。デベロッパーポータルが複数のサービスを提供するように設定するためのメインタスクは、API 利用者がサブスクライブするサービスを選択できるようにするページを作成することです。

詳細は、[マルチサービスへのサインアップ](#) を参照してください。

- デベロッパーポータルのページまたはページの一部を、指定する API 利用者にもみ表示するようにマークする。  
デベロッパーポータルの一部を、特定の API 利用者グループからしかアクセスできないように設定する必要がある場合があります。ページ、ページの一部、またはメニュー項目 (これらは通常セクションに対応します) へのアクセスを制限できます。

セクションへのアクセスを制限する簡単な方法は、各セクションを API 利用者の論理グループにマッピングすることです。たとえば、パートナーである API 利用者がいるとします。partner という名前のグループを作成し、そのグループにのみ特定のセクションへのアクセス権を付与できます。

状態の変更に応じて、API 利用者に制限のあるコンテンツへのアクセスを提供できます。たとえば、API 利用者が新しいアプリケーションプランにアップグレードすると、追加のページがその API 利用者に自動的に表示されるようになります。

アクセスを制限する別の方法は、特定のコンテンツを表示するために API 利用者にログインを要求することです。

詳細は [コンテンツの制限](#) を参照してください。

- Webhook を実装する。  
Webhook を使用すると、3scale をバックオフィスのワークフローと密接に連携させることができます。3scale システムで指定のイベントが発生すると、Webhook メッセージによりバックオフィスのアプリケーションに通知が届きます。その後、アプリケーションはそのデータ (新規アカウントに関する情報など) を使用して、デベロッパーポータルに反映することができます。

詳細は [Webhooks](#) を参照してください。

### 1.9.2.8 よりも前の 3SCALE リリースで使用されるデベロッパーポータルの更新

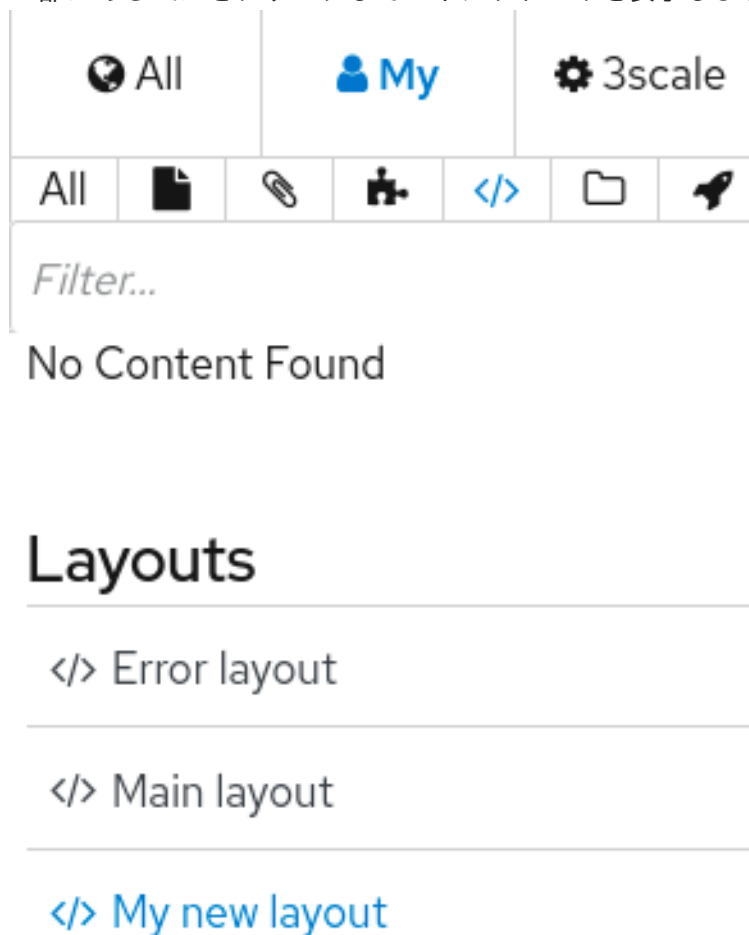
3scale 2.8 で、外部アセットがコンテンツ配信ネットワーク (CDN) から 3scale コードベースに移行されました。したがって、3scale 2.8 以降、`cdn_asset` Liquid タグを使用してネイティブのデベロッパーポータルが作成されます。2.8 よりも前の 3scale リリースからアップグレードする場合は、`cdn_asset` タグが付いた新しいアセットを使用するようにデベロッパーポータルを更新する必要があります。このタグの使用により、外部の Web サイトからアセットをダウンロードするための依存関係がなくなりました。

#### 前提条件

- 3scale 2.8 以降のインストール
- 3scale 2.7 以前で作成したデベロッパーポータル

#### 手順

1. 3scale 管理ポータルで **Audience > Developer Portal > Content** を選択して、リソース階層の上部にある `</>` をクリックしてレイアウトのみを表示します。



2. **Layouts** で **Main layout** をクリックします。
3. メインレイアウトのコードエディターで、**17** 行目またはその前後で、

```

{{ '//netdna.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.css' |
stylesheet_link_tag }}

```

上記を以下のように置き換えます。

```

{% cdn_asset /font-awesome/4.3.0/css/font-awesome.css %}

```

4. **19** 行目またはその前後で、

```

{{ '//ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js' | javascript_include_tag }}

```

上記を以下のように置き換えます。

```

{% cdn_asset /jquery/1.7.1/jquery.min.js %}

```

5. スクロールダウンして **Publish** をクリックします。

6. リソース階層の上部に移動し、 をクリックしてパーシャルを表示します。

7. **Partials** で **stats/chart** をクリックします。

8. `stats/chart` のコードエディターで、**3** 行目またはその前後で、

```
  {{ '//ajax.googleapis.com/ajax/libs/jqueryui/1.11.4/themes/ui-lightness/jquery-ui.css' |  
  stylesheet_link_tag }}
```

上記を以下のように置き換えます。

```
  {% cdn_asset /jquery-ui/1.11.4/jquery-ui.css %}
```

9. スクロールダウンして **Publish** をクリックします。
10. オプション:メインレイアウトに基づいてファイルを作成した場合や、`stats/chart` パーシャルに基づいてファイルを作成した場合は、同様にこれらのファイルを更新する必要があります。

## 関連情報

- 3scale Liquid リファレンスの [Liquid ドロップ](#)、[タグおよびフィルター](#)



## 第2章 カスタムのサインアップフォームフィールド

ここでは、カスタムのサインアップフィールドを追加する方法およびこの機能に関するさまざまなオプションについて説明します。

デフォルトの 3scale では、共通的に使用されるフィールドはユーザー/アカウント/アプリケーションのサインアップで設定します。これらの共通的なデフォルトフィールドに、独自のカスタムフィールドを追加する必要が生じる場合があります。

管理ポータルで **Audience > Accounts > Field Definitions**の順に移動し、ここでデフォルトのフォームフィールドを表示し、新規フィールドを定義することができます。

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

### Fields Definitions

Here you can manage all the information you gather from your users. You can add new fields and change the existing ones; making them Hidden, Read Only, or Required. You can change the text your users see when viewing or entering data (shown here between quotes). Drag and drop the fields to set the order in which they will be shown.

**Account** ➕ Create

✚ org_name	"Organization/Group Name"	Required <span>✎ Edit</span>
------------	---------------------------	------------------------------

**User** ➕ Create


✚ username	"Username"	Required <span>✎ Edit</span>
✚ email	"Email"	Required <span>✎ Edit</span>

**Application** ➕ Create

✚ name	"Name"	Required <span>✎ Edit</span>
✚ description	"Description"	Required <span>✎ Edit</span>

新しいアカウント/ユーザーのサインアップページは、実際には最初の2つのセクションを統合したものです。アカウントフィールドが上部に表示され、その後にユーザーフィールドが、そしてパスワードフィールド (設定の必要はない) が続きます。



 SIGN UP

ORGANIZATION/GROUP NAME

USERNAME

EMAIL

PASSWORD

PASSWORD CONFIRMATION

By signing up you agree to the following Legal Terms and Conditions ([show](#))

[Sign up](#)

新たにフィールドを3つ追加してみます。2つをユーザーサインアップセクションに、1つをアカウントセクションに追加します。Create をクリックして以下の新規フィールド定義を追加し、フィールド定義を作成します。Required チェックボックスを選択すると、当然、サインアップフォームで必須になります。フィールドを非表示や読み取り専用にするオプションも可能です。たとえば、新しいサインアップページに、デフォルトでは空である `access_restricted_areas` のようなユーザーの目に触れさせる必要のないフィールドセットを設定する場合、非表示フィールドを追加することができます。管理者は、後からこれをユーザーごとに true に更新できます。ページロジックはこれを読み取り、表示する項目を判断することができます。読み取り専用フィールドの例としては、ページの読み込み時に JavaScript を使用して設定することのできる、ブラウザの位置情報が挙げられます。

## New Field definition for User

Add a field to store information about your developers on signup or at any other time. Make the fields Hidden, Read Only, Required. The label is the text developers will see when viewing or entering their data.

FIELD FROM SCRATCH OR BASED ON EXISTING FIELD

[new field]

FIELD DETAILS

Name

last\_name

The low level system name.

Label

Last Name

The field title your developers will see.

Required

Makes the field required for developers.

Hidden

Developers won't be able to see this field.

Read only

Developers won't be able to change this field.

Choices

Full time, Part time, Contract

Separate the predefined options for this field by commas or enter each option on a new line.

Create

次に、ユーザーサインアップフォームにドロップダウンを追加してみます。Employment type を追加するとします。選択したフィールドに、コンマ区切りの値 (Full time, Part time, Contract) を追加します。ドロップダウンに、これらの値が反映されます。

FIELD FROM SCRATCH OR BASED ON EXISTING FIELD

[new field]

FIELD DETAILS

Name

employment\_type

The low level system name.

Label

Employment Type

The field title your developers will see.

 Required

Makes the field required for developers.

 Hidden

Developers won't be able to see this field.

 Read only

Developers won't be able to change this field.

Choices

Full time, Part time, Contract

Separate the predefined options for this field by commas or enter each option on a new line.

Create

ここで、事前定義済みのフィールドをアカウントに追加します。通常、追加するフィールドにはシステム機能がありません。フィールドは後からアクセス可能なデータを保持するだけです。(restricted contentを参照してください。)

通常どおりにフィールドを作成します。次に、Nameの上のドロップダウンリストで、po\_numberを選択します。このフィールドを使用すると、この開発者アカウントに送付される3scale生成の請求書にPO番号が表示されます。システムの生成するフィールドは、いつでも管理者が上書きすることができます。PO numberのような名前を付けて、フィールドを作成します。

## New Field definition for Account

Add a field to store information about your developers on signup or at any other time. Make the fields Hidden, Read Only, Required. The label is the text developers will see when viewing or entering their data.

FIELD FROM SCRATCH OR BASED ON EXISTING FIELD

✓ [new field]  
org\_legaladdress  
org\_legaladdress\_cont  
telephone\_number  
vat\_code  
vat\_rate  
fiscal\_code  
state\_region  
city  
country  
zip  
primary\_business  
business\_category  
po\_number  
billing\_address

Required

Makes the field required for developers.

Hidden

Developers won't be able to see this field.

Read only


Developers won't be able to change this field.

Choices

Separate the predefined options for this field by commas or enter each option on a new line.

Create

次に作業内容を確認します。ユーザーセクションに、自由記述の姓フィールドと、雇用タイプのドロップダウンが追加されていることを確認できます。同様に自由記述のPO番号のシステムフィールドが、アカウントセクションに追加されています。

 SIGN UP

ORGANIZATION/GROUP NAME

PO NUMBER

USERNAME

EMAIL

LAST NAME

EMPLOYMENT TYPE

PASSWORD

PASSWORD CONFIRMATION

By signing up you agree to the following Legal Terms and Conditions ([show](#))

これで 3scale ActiveDocs を使用してこれらのカスタムフィールドを設定できます (例: **Application Create**)。

## 第3章 サインアップフローの設定

本セクションでは、サインアップワークフローを調整するために行う設定について説明します。

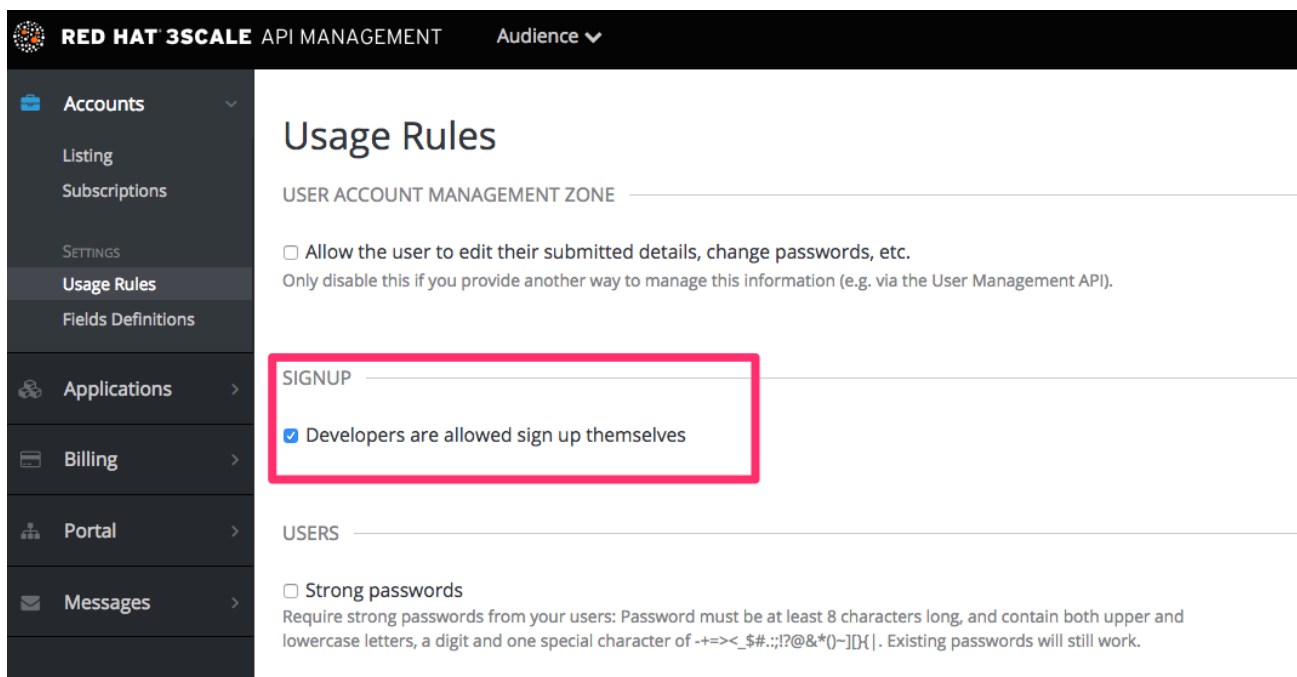
サインアップワークフローは、デベロッパーポータルを通じて提供する開発者体験の最も重要な部分です。このプロセスは、完全自動のセルフサービスから、逆に誰が何にアクセスできるかを完全に管理するものまで、さまざまな粒度に対応しています。

3scale プラットフォームでは、アカウント (オプション)、サービス (オプション)、およびアプリケーションプランの組み合わせを使用して API をモデル化できます。これらの各プランについて、API プロバイダーの承認を要するかどうかを制御できます。それぞれについて、デフォルトを提供するか、あるいは開発者が次の手順に進んで選択する必要があるかも決定します。

最大限の自動化とセルフサービスを許容する極端なケースでは、すべての承認ステップを削除し可能なデフォルトプランをすべて有効にします。これにより、サインアップ直後にキーを発行して、API へのアクセスを提供することができます。

### 3.1. すべての承認ステップの削除

承認ステップを削除するには、**Audience > Accounts > Usage Rules**の順に移動して、**SIGNUP** セクションで **Developers are allowed to sign up themselves** オプションのチェックボックスを選択します。



The screenshot shows the 'Usage Rules' configuration page in the Red Hat 3scale API Management interface. The left sidebar contains navigation options: Accounts, Applications, Billing, Portal, and Messages. The main content area is titled 'Usage Rules' and is divided into sections: 'USER ACCOUNT MANAGEMENT ZONE' and 'USERS'. In the 'SIGNUP' section, the checkbox 'Developers are allowed sign up themselves' is checked and highlighted with a red box. Below it, the 'USERS' section has an unchecked checkbox for 'Strong passwords'.

オプションとして、アカウントプランとサービスプランを有効にしている場合は、どちらのケースでも、ページをスクロールダウンして **Change the plan directly** オプションのチェックボックスを選択します。

RED HAT 3SCALE API MANAGEMENT Audience ▾

Accounts ▾

- Listing
- Account Plans
- Subscriptions

SETTINGS

- Usage Rules
- Fields Definitions

ADVANCED PLANS

Account Plans  
Only consider using Account Plans if Application Plans don't suit your use case - for example if you offer additional non-API services.

ACCOUNT PLANS CHANGING

Request a plan change

Change the plan directly

## 3.2. すべてのデフォルトプランの有効化

### アプリケーションプラン

RED HAT 3SCALE API MANAGEMENT API: Echo API ▾

Overview

Analytics

Applications ▾

- Listing
- Application Plans

Subscriptions

ActiveDocs

Integration

### Application Plans

Application Plans establish the rules (limits, pricing, features) for using your API; every developer's application accessing your API will be accessing it within the constraints of an Application Plan. From a business perspective, Application Plans allow you to target different audiences by using multiple plans (i.e. 'basic', 'pro', 'premium') with different sets of rules.

Basic  
 Unlimited  
 Plan A  
 Plan B

Name	Applications	State				
Basic	3	published	Hide	Copy	Delete	
Unlimited	4	published	Hide	Copy	Delete	
Plan A	0	published	Hide	Copy	Delete	
Plan B	0	hidden	Publish	Copy	Delete	

[Create Application Plan](#)

オプションとして、アカウントプランとサービスプランを有効にしている場合は、そのデフォルトプランも選択します。

### アカウントプラン (オプション)

RED HAT 3SCALE API MANAGEMENT Audience ▾

Accounts ▾

- Listing
- Account Plans
- Subscriptions

SETTINGS

- Usage Rules
- Fields Definitions

Applications

Billing

Developer Portal

### Account plans

Account plans create "tiers" of usage within the developer portal, allowing you to distinguish between grades of support, content and other services partners at different levels receive.

Default Plan  
Default  
Default account plan (if any) is contracted automatically on sign up.

[Edit the details of the plan from this link](#)

Name	Accounts	State			
Default	5	hidden	Publish	Copy	Delete

[Create Account Plan](#)

### サービスプラン (オプション)

RED HAT 3SCALE API MANAGEMENT API: Echo API

## Service Plans

Service plans allow you to define grades of service for each of the services (APIs) available through your developer portal. The plans allow you to define pricing per service and features available.

Default Plan  
  
 Default service plan (if any) is contracted automatically on sign up.

[Click on this link to edit features, etc.](#)

Name	Subscriptions	State			
Default	6	published	Hide	Copy	Delete

[Create Service Plan](#)

### 3.3. ワークフローのテスト

必要な設定変更を行ったら、デベロッパーポータルに移動して新しい開発者としてサインアップを試み、結果をテストします。完全な API のワークフローが得られるように、テストを行い必要な調整を加えます。ワークフローが満足できるものになったら、メール通知を確認して、開発者に適切な情報が提供されるようにします。

RED HAT 3SCALE API MANAGEMENT Audience

## Email Templates

Name	Description	
Buyer Account approved	After provider approves sign up, notification for buyer	Edit
Buyer Account confirmed	Buyer Account confirmed	Override
Credit card expired notification for buyer	Credit card expired notification for buyer	Override
Buyer account rejected	Buyer account rejected	Override
Alert notification for buyer (< 100%)	Alert notification for buyer when near 100% threshold	Override
Alert messenger limit alert for provider of master	No description.	Override
Alert notification for buyer (>= 100%)	Alert notification for buyer when over 100% threshold	Override



## 第4章 マルチサービスへのサインアップ

本セクションを完了すると、マルチサービスへのサインアップページを作成およびカスタマイズする手順を理解することができます。

マルチサービスの機能を使用している場合は、顧客がさまざまなサービスにサブスクライブできるようにサインアップ手順をカスタマイズすることができます。

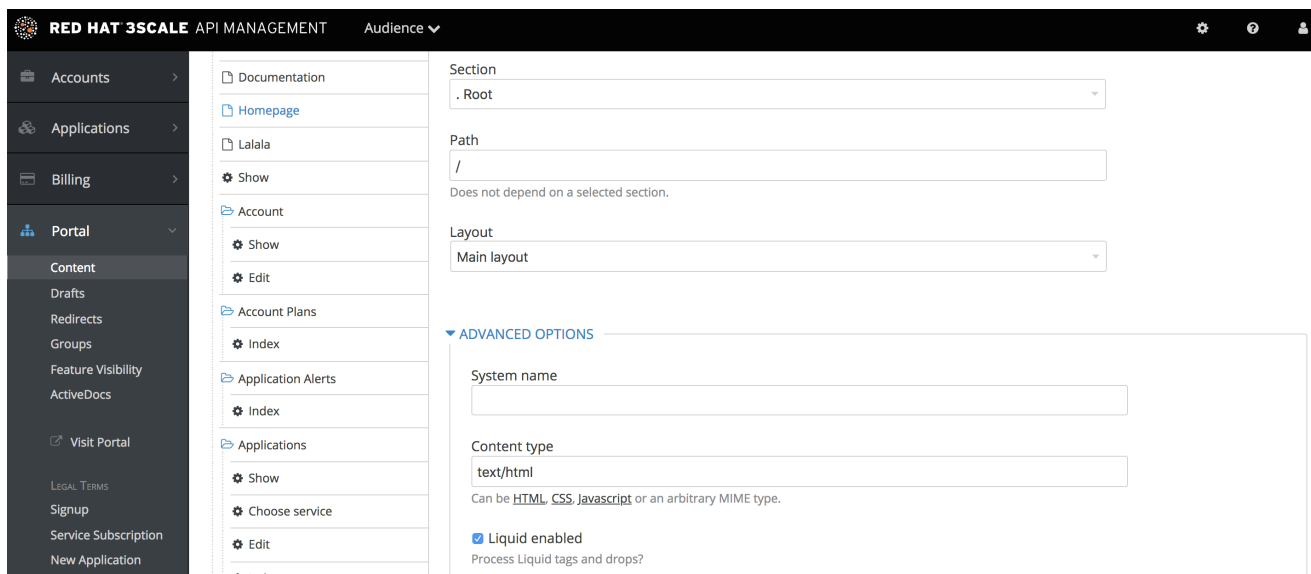
### 4.1. 前提条件

レイアウトとページ作成の手順、ならびに Liquid フォーマットタグの基本に関する知識が必要です。Liquid タグの詳細については、[Liquid リファレンス](#) を参照してください。マルチサービス機能をご自分のアカウントでも有効にする必要があります (Pro 以上のプランで利用可能)。

[サインアップワークフロー](#) に関する手順を読み理解することを強く推奨します。こうすることで、すべての設定を完了し、その仕組みを理解することができます。

### 4.2. はじめに

新しいレイアウトの作成からプロセスを開始します。このレイアウトは、マルチサービスへのサインアップページのテンプレートとして機能します。CMS システムの Layouts セクションに移動し、新しいレイアウトを作成します。他のレイアウトと簡単に区別できるようにするため、これを **multipleservicesignup** と呼ぶことにします。エディターで、標準レイアウト (home または main layout など) の一般構造を貼り付けます。次に、必要ではないもの (コンテナー、サイドバー、その他のボックスなど) をすべて削除します。



The screenshot shows the Red Hat 3Scale API Management interface. The top navigation bar includes 'RED HAT 3SCALE API MANAGEMENT' and 'Audience'. The left sidebar contains a menu with categories like 'Accounts', 'Applications', 'Billing', 'Portal', 'Content', 'Drafts', 'Redirects', 'Groups', 'Feature Visibility', 'ActiveDocs', 'Visit Portal', 'LEGAL TERMS', 'Signup', 'Service Subscription', and 'New Application'. The main content area is divided into three columns. The first column shows a tree view of the layout structure with items like 'Documentation', 'Homepage', 'Lalala', 'Show', 'Account', 'Account Plans', 'Application Alerts', 'Applications', and 'Index'. The second column shows configuration fields for the selected item: 'Section' (set to '. Root'), 'Path' (set to '/'), and 'Layout' (set to 'Main layout'). The third column shows 'ADVANCED OPTIONS' with fields for 'System name', 'Content type' (set to 'text/html'), and a checked checkbox for 'Liquid enabled'.

レイアウトのベースを作成したら、サインアップ用コードのカスタマイズに進みます。

### 4.3. マルチサービスへのサインアップ

#### 4.3.1. サービスに関する情報の取得

適切なサインアップリンクを構築する必要があるサービスについて、その情報をすべて取得するためには、サービスオブジェクトをループスルーしなければなりません。サービスは、モデルオブジェクトの一部です。

```
{% for service in provider.services %}
.
.
.
{% endfor %}
```

### 4.3.2. サインアップカラムの設定

すでにレイアウトとサービスオブジェクトにアクセスするループがあります。次に、サービスとサインアップリンクに関する情報の表示方法を決定します。たとえば、カラムに分けて、サービスの説明とサインアップリンク (カラム下部) を表示します。すべてのカラムは `service-column` クラスの `div` ボックスに対応し、ここに必要なすべての情報が収められます。

```
{% for service in provider.services %}
<div class="service-column">
  <p>{{ service.name }}</p>
  <p>{{ service.description }}</p>
  .
  .
  .
</div>
{% endfor %}
```

コンテナ内部はカスタムの説明フィールドとして機能します。 `service.name` はサービスの名前で、ここではコンテナの名前になります。

### 4.3.3. サブスクリプションの設定

次は、カスタムのサービスサインアップの主要部分です。サインアップリンクを作成するため、サインアップ URL とサービス ID を抽出します。URL のオブジェクトからサインアップ URL を、ループで繰り返すサービスオブジェクトからサービス ID を、それぞれ取得します。最終的なリンクコードは以下のようになります。

```
<a href="{{ urls.signup }}?{{ service | toparam }}">Signup to {{ service.name }}</a>
```

いくつかのサービスについて、ユーザーがすでにサインアップ済みであることも考慮する必要があります。確認するための条件ブロックを作成します。

```
{% unless service.subscribed? %}
  <a href="{{ urls.signup }}?{{ service | toparam }}">Signup to {{ service.name }}</a>
{% endunless %}
```

これを使用して、最終的なコードを生成できます。

```
{% for service in provider.services %}
<div class="service-column">
  <p>{{ service.name }}</p>
  <p>{{ service.description }}</p>
  {% unless service.subscribed? %}
    <a href="{{ urls.signup }}?{{ service | to_param }}">Signup to {{ service.name }}</a>
  {% endunless %}
</div>
{% endfor %}
```

#### 4.3.4. スタイリング

対象のサービスの数に応じて、生成したマークアップに最終的な仕上げを行います。この例ではサービスが2つあるので、`service-column` div の CSS コードは以下のようになります。

```
.service-column {  
  float: left;  
  margin-left: 10%;  
  width: 45%;  
}  
.service-column:first-child {  
  margin-left: 0;  
}
```

この例では、パーセントベースのレイアウトを使用して、含まれる div の大きさを元にカラムの幅を動的に割り当てています。

これで、正常に動作し、見た目の整った複数サービスのサブスクリプションページができたはずです。お疲れさまでした。

カラムを特定の順序で表示する場合は、サービス名または利用可能なその他の値を条件とする条件式 (if/else/case) を使ってみてください。

## 第5章 デベロッパーポータルの認証

デベロッパーポータルへのアクセスを設定するには、以下の手順に従います。

本章では、開発者のサインアップまたはサインインを許可するためにデベロッパーポータルで利用可能なさまざまなタイプの認証について、有効および無効にする方法を説明します。

現時点で、デベロッパーポータルへの認証に関して、3scale はさまざまな方法をサポートしています。それらを以降のセクションで説明します。

1. [ユーザー名/メールアドレスおよびパスワードによる認証](#)
2. [GitHub による認証](#)
3. [Auth0 による認証](#)
4. [Red Hat Single Sign-On による認証](#)

デフォルトでは、デベロッパーポータルで有効にできる認証方法は1つだけですが、3scale.net でサインアップした場合は2つになります。

- ユーザー名/メールアドレスおよびパスワードによる認証
- GitHub による認証 (3scale GitHub アプリケーションを使用): デフォルトでは 3scale.net でサインアップした場合にのみ有効

### 注記

2015年12月14日より前に作成された古い3scaleアカウントの場合、GitHub認証およびAuth0認証を有効にするためには、また別の手順に従わなければならない可能性があります。

これに該当する場合、ログインおよびサインアップ用のフォームでこの機能を有効にするためには、以下のコードスニペットを両方のテンプレートに追加する必要があります。

```
{% include 'login/sso' %}
```

## 5.1. ユーザー名/メールアドレスおよびパスワードによる認証の有効化と無効化

デベロッパーポータルでは、デフォルトでユーザー名/メールアドレスおよびパスワードによる認証が有効です。これは開発者がアカウントを作成してログインするための標準的な方法なので、通常は、ここを変更することはありません。

ただし、まれにですが、この認証タイプを削除する必要があります。そのためには、下記のスクリーンショットで示すように **Login > New** テンプレートを編集します。

```

1 <div class="row">
2   <div class="col-md-9">
3     <div class="panel panel-default">
4       <div class="panel-heading">
5         <i class="fa fa-user"></i>
6         Sign in
7       </div>
8       <div class="panel-body">
9         {% include 'login/sso' %}
10
11         {% comment %}
12
13         {% form 'login_form', class: 'form-horizontal' %}
14           {% include 'login/cas' %}
15           {% include 'login/rainrain' %}
16
17           <fieldset>
18             <div class="form-group" id="session_username_input">
19               <label for="session_username" class="control-label col-md-4">Username or Email</label>
20               <div class="col-md-6">
21                 <input id="session_username" name="username" tabindex="1" autofocus="autofocus"
22                   type="text"
23                   class="form-control">
24               </div>
25             </div>
26           </div>
27           <div class="form-group" id="session_password_input">
28             <label for="session_password" class="control-label col-md-4">Password</label>
29             <div class="col-md-6">
30               <input id="session_password" name="password" tabindex="2"
31                 type="password"
32                 class="form-control">
33             </div>
34           </div>
35           <input name="remember_me" type="hidden" value="1">
36         </fieldset>
37         <fieldset>
38           <div class="form-group">
39             <div class="col-md-10">
40               <input name="commit" type="submit" value="Sign in" class="btn btn-success btn-lg pull-
41 right">
42             </div>
43           </div>
44         </fieldset>
45       </div>
46       {% endform %}
47       {% endcomment %}
48     </div>
49   </div>
50   <div class="panel-footer">
51     <a href="{{ urls.forgot_password }}">Forgot password?</a>
52
53     {% if provider.signups_enabled? %}
54     | <a href="{{ urls.signup }}" class="link">Sign up</a>
55     {% endif %}
56   </div>
57 </div>
58 </div>
59 </div>
60 </div>
61

```

sso authentication snippet

wrap the form with these comment tags

デベロッパーポータルにユーザー名/メールアドレスおよびパスワードによる認証を追加して元の状態に戻す必要がある場合は、前のステップで追加した Liquid コメントタグを削除するだけです。

## 5.2. GITHUB による認証の有効化および無効化

専用の GitHub アプリケーションを有効にするには、まずアプリケーションを作成し、対応するクレデンシャルを取得する必要があります。

GitHub による認証を設定するには、2 種類の方法があります。

- 3scale GitHub アプリケーションを使用する (ホスト型 3scale アカウントではデフォルトで有効)
- 専用の GitHub アプリケーションを使用する (オンプレミス型のインストール環境用)

このデフォルト設定に変更を加える場合、3scale 管理ポータルで **Audience > Developer Portal > SSO Integrations** の順に移動すると、以下のような画面が表示されます。

The screenshot shows the 'Single Sign-on Integrations' page in the 3scale API Management interface. The page has a dark sidebar on the left with a menu containing items like 'Content', 'Drafts', 'Redirects', 'Groups', 'Feature Visibility', 'ActiveDocs', 'Visit Portal', 'LEGAL TERMS', 'Signup', 'Service Subscription', 'New Application', 'SETTINGS', 'Domains & Access', 'Spam Protection', 'SSO Integrations', and 'Forum Settings'. The main content area is titled 'Single Sign-on Integrations' and contains a table with the following data:

Integration	State
<a href="#">GitHub</a>	Hidden
<a href="#">★ Auth0</a>	Hidden
<a href="#">Red Hat Single Sign-On</a>	Hidden

GitHub をクリックして設定画面にアクセスします。



[Edit](#)

Published:

Branding: 3scale branded

Authentication Flow: Test

この画面では、以下の操作を行うことができます。

1. デベロッパーポータルでの GitHub 認証を利用可能にしたり利用不可にしたりする。その操作は、Published チェックボックスを選択または選択を解除するだけです。
2. 3scale ブランドの GitHub アプリケーションを選択する、または専用の GitHub アプリケーションを追加する。3scale GitHub アプリケーションはデフォルトで有効です (公開されています)。Edit をクリックし、GitHub で作成した OAuth アプリケーションの詳細 (Client と Client secret) を入力することで、専用の GitHub アプリケーションを設定できます。専用の GitHub アプリケーションでインテグレーションを適切に機能させるためには、custom branded オプションを切り替えた後に表示される Callback URL を使用して GitHub アプリケーションの承認コールバック URL を設定しなければならないことに留意してください (例: <https://yourdomain.3scale.net/auth/github/callback>)。
3. 設定した認証フローが想定どおりに機能することをテストする。

## 5.3. AUTH0 による認証の有効化および無効化

### 5.3.1. 注記

この機能は、Enterprise プランでのみ利用可能です。

開発者が Auth0 を使用して認証できるようにするためには、まず有効な Auth0 サブスクリプションが必要です。

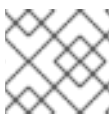
Auth0 による認証をデフォルトで有効にすることはできません。デベロッパーポータルへのアクセスを管理するために 3scale と Auth0 アカウントを組み合わせる場合は、以下の手順に従ってその設定を行うことができます。

3scale 管理ポータルで **Audience > Developer Portal > SSO Integrations**の順に移動し、**Auth0** をクリックします。

The screenshot shows the 'New Auth0 Authentication Provider' configuration page in the 3scale API Management console. The page has a dark sidebar on the left with navigation options like 'Content', 'Drafts', 'Redirects', 'Groups', 'Feature Visibility', 'ActiveDocs', 'Visit Portal', 'LEGAL TERMS', 'Signup', 'Service Subscription', 'New Application', 'SETTINGS', 'Domains & Access', 'Spam Protection', 'SSO Integrations', and 'Forum Settings'. The main content area has a title 'New Auth0 Authentication Provider' and three input fields: 'Client ID', 'Client Secret', and 'Site'. Below the 'Site' field is an example: 'Ex: https://XXXXX.auth0.com'. There is a checkbox labeled 'Published' with the text 'Display on Developer Portal' below it. At the bottom right, there is a blue button labeled 'Create Auth0'.

この設定画面で、Auth0 アカウントの詳細情報を追加する必要があります。クライアント ID、クライアントシークレット、およびサイトを入力したら、Published チェックボックスを選択して **Create Auth0** をクリックし、デベロッパーポータルで利用できるようにします。

## 5.4. RED HAT SINGLE SIGN-ON による認証の有効化および無効化



### 注記

この機能は、Enterprise プランでのみ利用可能です。

Red Hat Single Sign-On (RH-SSO) は、統合サインオンソリューション (SSO) です。3scale と組み合わせて使用すると、利用可能な任意の RH-SSO アイデンティティブローカー機能とユーザーフェデレーションオプションを使用して、開発者を認証することができます。

3scale との互換性がある Red Hat Single Sign-On のバージョンについての情報は、[サポートされる設定に関するページ](#)を参照してください。


### 5.4.1. 操作を始める前に





Red Hat Single Sign-On を 3scale と統合する前に、動作状態にある Red Hat Single Sign-On インスタンスが必要です。[RH-SSO 7.2 のインストール](#) に関する手順については、Red Hat Single Sign-On のドキュメントを参照してください。



## 5.4.2. デベロッパーポータルを認証するための RH SSO の設定

Red Hat Single Sign-On を設定するには、以下の手順を実施します。


1. [Red Hat Single Sign-On のドキュメント](#) に記載の手順に従って、レلمを作成します。
2. **Clients** に移動して **Create** をクリックし、クライアントを追加します。
3. 以下のフィールドと値を考慮してフォームに入力します。
  - **Client ID**: クライアントの希望の名前を入力します。
  - **Enabled**: **ON** に切り替えます。
  - **Consent Required**: **OFF** に切り替えます。
  - **Client Protocol**: **openid-connect** を選択します。
  - **Access Type**: **confidential** を選択します。
  - **Standard Flow Enabled**: **ON** に切り替えます。
  - **Root URL**: 3scale 管理ポータルの URL を入力します。これは、デベロッパーポータルへのログインに使用する URL アドレスでなければなりません (例: <https://yourdomain.3scale.net> またはカスタムの URL)。
  - **Valid Redirect URLs**: /\* を付けて再度デベロッパーポータルを入力します (例: [https://yourdomain.3scale.net/\\*](https://yourdomain.3scale.net/*))。その他のパラメーターはすべて空のままにするか、**OFF** に切り替える必要があります。
4. 以下の手順によりクライアントシークレットを取得します。
  - 前のステップで作成したクライアントに移動します。
  - **Credentials** タブをクリックします。
  - **Client Authenticator** フィールドで、**Client Id and Secret** を選択します。

Account 

Settings **Credentials** Roles Mappers  Scope  Revocation Sessions  Offline Access 

Client Authenticator  Client Id and Secret 

Secret

Registration access token 

5. **email\_verified** マッパーを設定します。3scale では、ユーザーデータの **email\_verified** 要求が **true** に設定されている必要があります。Email Verifiedユーザー属性を **email\_verified** 要求にマッピングするには、以下の手順を実施します。
  - クライアントの **Mappers** タブに移動します。
  - **Add Builtin** をクリックします。



Name	Category	Type	Actions
full name	Token mapper	User's full name	Edit Delete
given name	Token mapper	User Property	Edit Delete
email	Token mapper	User Property	Edit Delete
username	Token mapper	User Property	Edit Delete
family name	Token mapper	User Property	Edit Delete

- **email verified** オプションのチェックボックスを選択し、**Add selected** をクリックして変更を保存します。

Name	Category	Type	Add
email verified	Token mapper	User Property	<input checked="" type="checkbox"/>
locale	Token mapper	User Attribute	<input type="checkbox"/>
address	Token mapper	User Address	<input type="checkbox"/>
gss delegation credential	Token mapper	User Session Note	<input type="checkbox"/>

Red Hat Single Sign-On ローカルデータベースでユーザーを管理する場合は、必ずユーザーの **Email Verified** 属性を **ON** に設定してください。

[ユーザーフェデレーション](#) を使用する場合は、3scale SSO インテグレーション用に前のステップで作成したクライアントで、トークン名を **email\_verified** とし要求の値を **true** に設定して、ハードコーディングされた要求を設定することができます。

6. オプションとして、**org\_name** マッパーを設定します。  
ユーザーは 3scale にサインアップするとき、サインアップフォームに組織名の値を入力するよう求められます。デベロッパーポータルでのサインアップフォームの入力を不要にして、ユーザーが Red Hat Single Sign-On によるサインアップを意識しないようにするには、さらに **org\_name** マッパーを設定する必要があります。
  - クライアントの **Mappers** タブに移動します。
  - **Create** をクリックします。
  - 以下のようにマッパーのパラメーターを入力します。
    - **Name:** 希望する任意の名前を入力します (例: **org\_name**)。
    - **Consent Required:** **OFF** に切り替えます。
    - **Mapper Type:** **User Attribute** を選択します。
    - **User Attribute:** **org\_name** を入力します。
    - **Token Claim Name:** **org\_name** を入力します。
    - **Claim JSON Type:** **String** を選択します。
    - **Add to ID token:** **ON** に切り替えます。
    - **Add to access token:** **ON** に切り替えます。

- **Add to userinfo: ON** に切り替えます。
- **Multivalued: OFF** に切り替えます。
- **Save** をクリックします。

## Create Protocol Mapper

Protocol	<input type="text" value="openid-connect"/>
Name	<input type="text" value="org_name"/>
Consent Required	<input type="checkbox"/> OFF
Mapper Type	<input type="text" value="User Attribute"/>
User Attribute	<input type="text" value="org_name"/>
Token Claim Name	<input type="text" value="org_name"/>
Claim JSON Type	<input type="text" value="String"/>
Add to ID token	<input checked="" type="checkbox"/> ON
Add to access token	<input checked="" type="checkbox"/> ON
Multivalued	<input type="checkbox"/> OFF
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

Red Hat Single Sign-On のユーザーに **org\_name** 属性があれば、3scale は自動的にアカウントを作成することができます。属性がないユーザーについては、アカウント作成の前に組織名を指定するよう求められます。あるいは、Red Hat Single Sign-On アカウントでサインインするすべてのユーザーについて、**Hardcoded claim** タイプのマッパーを作成して、組織名をハードコーディング値に設定することもできます。


7. インテグレーションをテストするには、ユーザーを追加する必要があります。そのためには、**Users** に移動し、**Add user** をクリックして必須フィールドに入力します。Red Hat Single Sign-On でユーザーを作成する場合、Email Verified 属性 (**email\_verified**) が **ON** に設定されている必要があることに留意してください。設定されていないと、ユーザーが 3scale でアクティブ化されません。


## Red Hat Single Sign-On のアイデンティティブローカーとしての使用

Red Hat Single Sign-On をアイデンティティブローカーとして使用することや、外部データベースをフェデレーションするように設定することができます。これらの設定方法については、Red Hat Single Sign-On のドキュメントで、[アイデンティティブローカー設定](#) および [ユーザーフェデレーション](#) に関する章を参照してください。

アイデンティティブローカーとして Red Hat Single Sign-On の使用を選択し、さらに開発者が RH-SSO と 3scale のアカウント作成手順の両方をスキップできるようにすることを希望する場合は、下記の設定を行うことを推奨します。以下の例では、アイデンティティプロバイダーとして GitHub を使用しています。

1. Red Hat Single Sign-On で、**アイデンティティプロバイダー** に GitHub を設定してから、**Mappers** というタブに移動し、**Create** をクリックします。

Organization Name 

ID	<input type="text"/>
Name * 	<input type="text" value="organization name"/>
Mapper Type 	<input type="text" value="Attribute Importer"/>
Social Profile JSON Field Path 	<input type="text" value="company"/>
User Attribute Name 	<input type="text" value="org_name"/>
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

- 名前を付けて、識別できるようにします。
- Mapper Type で Attribute Importer を選択します。
- Social Profile JSON Field Path に company を追加します。これは GitHub での属性の名前です。
- User Attribute Name に org\_name を追加します。これは Red Hat Single Sign-On での属性の呼び方です。



## 注記

Red Hat Single Sign-On では、必須フィールドとして姓および名に加えてメールアドレスの入力が必要です。3scale では、メールアドレス、ユーザー名、および組織名が必要です。したがって、組織名のマッパーを設定することに加え、ユーザーが両方のサインアップフォームをスキップできるよう、以下のことを確認してください。

- IdP アカウントで、姓と名が設定されている。
- IdP アカウントで、メールアドレスにアクセスできる。つまり、GitHub で、メールアドレスがプライベートに設定されている場合、共有されません。

## 5.4.3. デベロッパーポータルを認証するための 3scale の設定

API プロバイダーは、Red Hat Single Sign-On (RH-SSO) を使用したデベロッパーポータルの認証が可能になるように 3scale を設定します。



## 注記

RH-SSO を使用した認証は、デフォルトでは有効になっていません。RH-SSO は、エンタープライズ 3scale アカウントでのみ利用可能です。したがって、アカウントマネージャーに RH-SSO による認証を有効にするように依頼する必要があります。

## 前提条件

- エンタープライズ 3scale アカウントが、RH-SSO を有効にするように設定されている。
- [デベロッパーポータルを認証するために RH SSO を設定した後](#)、以下の詳細情報を取得している。
  - クライアント: RH-SSO でのクライアント名

- **クライアントシークレット**: RH-SSO でのクライアントシークレット
- **レルム**: レルム名および RH-SSO アカウントへの URL アドレス

## 手順

1. 3scale 管理ポータルで **Audience > Developer Portal > SSO Integrations**の順に選択します。
2. **Red Hat Single Sign-On**をクリックします。
3. 「[デベロッパーポータルを認証するための RH SSO の設定](#)」で設定した RH-SSO クライアントの詳細(クライアント、クライアントシークレット、およびレルム)を指定します。
4. 変更を保存するには、**Create Red Hat Single Sign-On**をクリックします。

## 第6章 デベロッパーポータル用の RED HAT SINGLE SIGN ON

Red Hat Single Sign On (RH SSO) を使用すると、独立した複数のシステムのアクセス制御を管理できます。本章の手順に従うと、ご自分のシステムにログインするユーザーは、再度ログインを求められることなく、3scale デベロッパーポータルに自動的にログインすることができます。

ここでは、Web サイトの既存のユーザークレデンシャルを使って、どのように 3scale デベロッパーポータルに自動的にログインするかを説明します。

この機能は、API 利用者の ID (ユーザー名およびパスワード) をすでに所有している API プロバイダー向けのもので (たとえば、API プロバイダーがアイデンティティプロバイダーでもあるケース)。

### 6.1. 3SCALE プラットフォームでのユーザーの作成

まず、API 利用者は、デベロッパーポータルにアカウントがなければなりません。Account Management API を使用してユーザーを 3scale にインポートすることも、手動で作成することもできます。3scale ActiveDocs (管理ポータルの **Documentation (右上隅の疑問符 (?) のアイコン)** → **3scale API Docs** セクションから利用可能) で Account Management API を検索します。

### 6.2. ログインリンクのリクエスト

ユーザーが存在すると、API リクエストコールを使用して、組み込み SSO トークンにより URL を生成できます。

```
curl -X POST -d "provider_key=YOUR_PROVIDER_KEY&username=USERNAME&expires_in=60"
https://YOUR_ADMIN_PORTAL.3scale.net/admin/api/sso_tokens.xml
```

この呼び出しには 2 つのパラメーターがあります。誰のトークンをリクエストするのかを指定する `username` と、トークンが有効である秒数 (デフォルトでは 10 分) の `expires_in` です。

ユーザーが正常にログインした後にリダイレクトされる場所と共に追加パラメーターの `redirect_url` を渡すこともできます。このパラメーターは、[パーセントエンコーディング](#) されている必要があります。XML レスポンスには、シークレットトークンが含まれる URL が含まれます。

```
<?xml version="1.0" encoding="UTF-8"?>
<sso_url>
https://YOUR_DEVELOPER_PORTAL/session/create?
expires_at=1365087501&token=Q0dNWGtjL2h2MnloR11yWmNwazVZY0NhenlabnBoRUNaNUlyWjZa
VG8wMnBGdVNHt0VGN1NUb3FRc1pwSnRrcIBZSTlwOUFWrkVTc3NuK1JTbjUrMEE9PS0tY1ZrOG
FldzFJNkxna1hrQzQyZ0NGQT09--712f2990ac9248ab4b8962be6467fb149b346000
</sso_url>
```

#### 注記

3scale ユーザーを識別するために `user_id` か `username` のいずれかを渡すことができます。通常、システムと 3scale ポータルの `username` は同じです。この場合、API プロバイダー側では何らかの追加情報を保存しておく必要がないので、`username` を使用することは容易なはずですが、ただし、URL について何らかのペアリングや機械処理が必要な場合は、`user_id` を使用の方が望ましいです。

### 6.3. 自動ログインによるユーザーのリダイレクト

レスポンスには、トークンが含まれる RH SSO ログイン URL が含まれます。

```
https://YOUR_DEVELOPER_PORTAL/session/create?  
expires_at=1365087501&token=Q0dNWGtjL2h2MnloR11yWmNwazVZY0NhenlabnBoRUNaNUlyWjZa  
VG8wMnBGdVNHt0VGN1NUb3FRc1pwSnRrcIBZSTlwOUFwRkVTc3NuK1JTbjUrMEE9PS0tY1ZrOG  
FldzFJNkxna1hrQzQyZ0NGQT09--712f2990ac9248ab4b8962be6467fb149b346000
```

URL には、3scale デベロッパーポータル SSO がログインを許可するのに必要なすべての情報が含まれます。これを Web に直接埋め込むことができます。ただし、URL はユーザーがクリックする前に期限切れになることがあるので、ページには、動的に新しい SSO URL をリクエストしてそこにリダイレクトする汎用リンクを用意することが推奨されます。このようにして、ユーザーをデベロッパーポータルにシームレスにログインさせます。



### 注記

URL のアドレスはエスケープ解除する必要があります。これをブラウザで手動で行う場合には、ブラウザで **&amp;** を **&** に置き換えるのを忘れないでください。また、トークンの % エンコーディングは、すべてその非エスケープ文字で置き換える必要があります。

## 第7章 コンテンツの制限

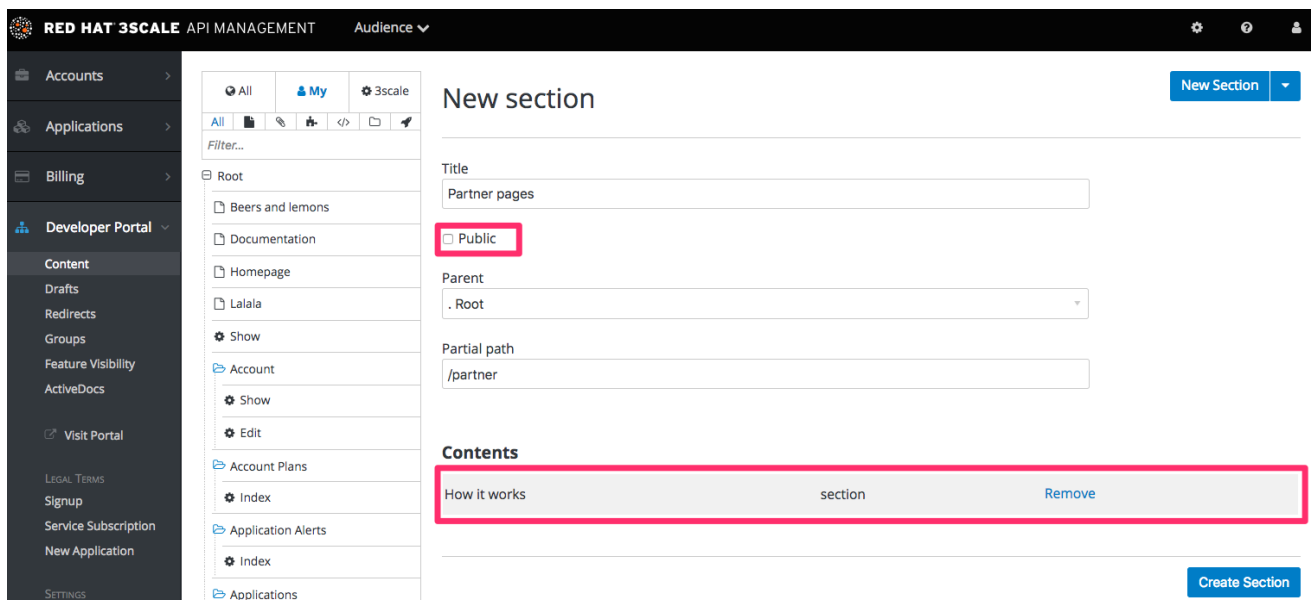
本章では、デベロッパーポータルコンテンツの一部のユーザーにしか見えないようにする方法を説明します。

デベロッパーポータルの一部のページについて、ページの一部またはあるメニュー項目を、特定の開発者グループだけがアクセスできるように設定しなければならない場合があります。いずれの場合も、以下で紹介する2つの手法により目的を達成することが可能です。

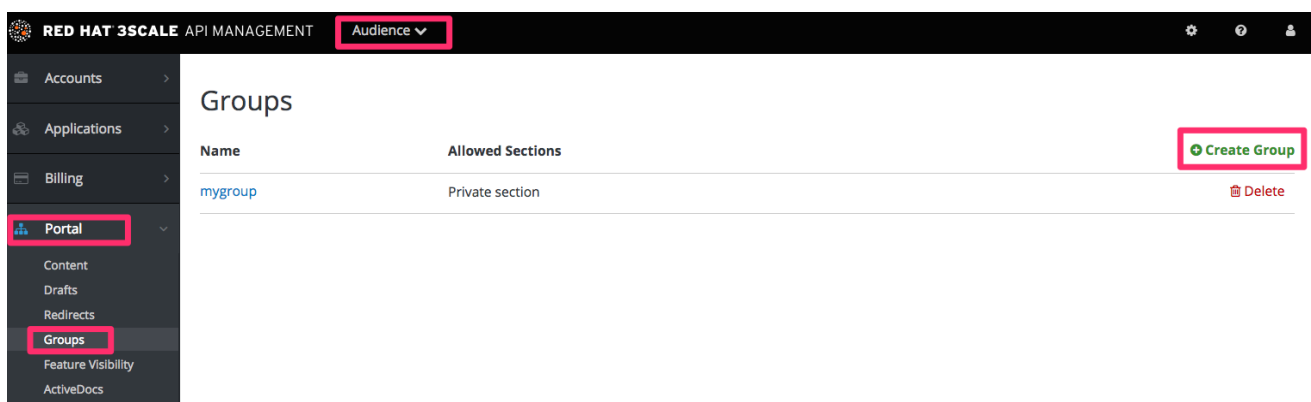
### 7.1. ページの制限

制限付きのセクションを作成する場合、各セクションをユーザーの論理グループにマッピングさせると便利です。以下の例では、partners という開発者グループがあると仮定します。

アクセスを制限するすべてのページまたはページのグループについて、デベロッパーポータルで新しいセクションを作成します。public ステータスフィールドのチェックボックスの選択を解除します。次に、このセクション内に必要なすべてのページをドラッグアンドドロップします。



グループを作成して、作成したセクションへのアクセス権限を付与します。



これで、このセクションへのアクセス権限をユーザーの一部に付与するときには、このグループに割り当てただけで済みます。そのためには、該当するアカウントの詳細ページから Group Permissions に移動します。移動したら、許可するセクションのチェックボックスを選択します。

RED HAT 3SCALE API MANAGEMENT Audience

Accounts Listing Account Plans Subscriptions SETTINGS Usage Rules Fields Definitions

Account 'Metro' 1 Application | 1 User | 0 Invitations | 0 Group Memberships | 0 Invoices | 1 Service Subscription

## Groups of 'Metro'

Groups

mygroup (Private section)

## 7.2. コンテンツブロックの制限

Liquid タグを使用すると、デベロッパーポータルを自由にカスタマイズすることができます。本章の用途に Liquid タグを使用して、条件に基づきページの一部を表示/非表示にします。3scale では、アカウント、アプリケーション、およびユーザーにカスタムフィールドを作成できます。これを利用して、API プロバイダーに役立つ情報を格納できます。ここでは、すべてのアカウントにアタッチされるカスタムフィールドを作成し、これを使ってそのアカウントがパートナーかそうでないかを識別します。**Audience > Account > Field Definitions**の順に移動して、このフィールドを作成することができます。Account セクションにフィールドを追加してそれを非表示として設定すると、サインアップページやポータルなどのページにも表示されなくなります。

RED HAT 3SCALE API MANAGEMENT Audience

## Fields Definitions

Here you can manage all the information you gather from your users. You can add new fields and change the existing ones; making them Hidden, Read Only, or Required. You can change the text your users see when viewing or entering data (shown here between quotes). Drag and drop the fields to set the order in which they will be shown.

Account

Field Name	Field Value	Field Status	Field Action
* org_name	"Organization/Group Name"	Required	Create Edit
* Partner	"partner"	Hidden	Edit

カスタムフィールドを設定すると、以下のスニペットに示すように条件でラップすることで、パートナーに特別なコンテンツを表示できるようになります。

```

{{ if current_account.extra_fields.partner == 'true' }}
  // content only accessible to partners
{{ endif }}

```

また、その方が適切であれば、逆のロジックを使用します。

```

{{ unless current_account.extra_fields.partner == 'true' }}
  // content forbidden for partners
{{ endunless }}

```

これ以降、この非表示コンテンツをユーザーに表示する場合には、該当ユーザーのアカウント詳細ページで partner フィールドに true と入力するだけです。

## 7.3. 追加フィールドの設定の自動化



開発者に制限付きコンテンツへのアクセス権限を付与する場合があります。たとえば、アプリケーションプランをアップグレードした場合などです。

Account Management API と [Webhook](#) を組み合わせて使用して、このプロセスを簡素化します。Account Management API は、管理ポータルにある 3scale ActiveDocs に含まれています。

1. ウィンドウの右上にある **Documentation** (疑問符 (?)) をクリックします。
2. **3scale API Docs** を選択します。
3. Webhook 要求によって送信されたメッセージをチェックして、制限付きコンテンツにアクセスする開発者の新しいプランを取得します。
4. 開発者の新しいプランに基づいて、**partner** フィールドを更新する API を呼び出すことで、プライベートコンテンツへのアクセス権限を付与することができます。

## 7.4. ユーザーログインの要求

コンテンツへのアクセスを制限する上述の 2 つの手法に加えて、ユーザーにログインを要求するという別の有用な手法があります。

これは、Liquid タグを使用して非常に簡単に実施することができます。ログインしたユーザーのみが利用できるコンテンツを、以下の条件内にラップするだけです。

```
{{ if current_user }}  
  // only visible if the user is logged in  
{{ endif }}
```

## 第8章 メールテンプレート

本章では、カスタムメールテンプレートを編集して保存する方法について説明します。

開発者とのあらゆる標準メール通信のコンテンツを完全にカスタマイズすることができるので、デベロッパーポータルに設定してあるワークフローに厳密にマッチさせることが可能です。

### 8.1. メールテンプレートのカスタマイズ

#### 8.1.1. メール設定の前に行うワークフローの定義

メールテンプレートオプションは多数ありますが、一部のテンプレートだけがご自分のワークフローに関連します。メールテンプレートの編集を始める前に、時間節約のため、ワークフローに問題がないことを確認します。こうすることで、実際に使用するテンプレートだけを編集することができます。

#### 8.1.2. ワークフローのテストおよび有効なメールテンプレートを識別

最終的なワークフローのドライランを実行し、考え得る状況(承認や拒否など)をすべて確実にテストします。次に、テスト用の開発者アカウントが受信する各メール通知を識別し、次のステップで何を編集すべきかを判断します。

#### 8.1.3. カスタムテンプレートの編集および保存

テンプレートを初めて編集する場合、実際にカスタムテンプレートを作成することになります。その後の編集で、変更を保存します。警告: バージョン管理はありません。変更を元に戻せるようにしたい場合は、ローカルコピーを作成することを推奨します。

電子メールの動的コンテンツに Liquid タグを使用できます。Liquid タグに変更を加える場合は特に、バックアップを作成することを推奨します。

RED HAT 3SCALE API MANAGEMENT Audience

## Edit template "Buyer Account approved"

You can use Liquid tags to set the email headers or disable sending. Read more in the [liquid documentation](#).

Subject  
use 3scale default

Bcc

Cc

Reply to

From  
use 3scale default

**Liquid tags for dynamic email customisation**

```

1 {% email %}{% do_not_send %}{% endemail %}
2
3 Dear {{ user.display_name }},
4
5 {{ provider.name }} has approved your signup for the {{ provider.name }} API.
6
7 You may now view and manage your app/API key at https://{{ provider.domain }}/admin/
8
9 If you have problems logging into the account please contact {{ provider.support_email }}.
10
11 Sincerely,
12 The {{ provider.name }} API Team
13

```

Disable Sending Snippet Save

### 8.1.4. ワークフロー内の全テンプレートについての反復

ワークフローで考え得るすべての状況をカバーするまで、同じ手順を実施します。

## 8.2. 補足情報

- メールテンプレートをカスタマイズする前に、[サインアップフロー](#)を完全に確定してテストしておくことが最善です。
- メールテンプレート内の Liquid タグを変更しようとする場合は、必ず [Liquid に関する参考ドキュメント](#) を十分に調べてください。

## 第9章 LIQUID: デベロッパーポータル

このセクションでは、マークアップのさまざまな要素、その要素のつながり、デベロッパーポータルでの使用方法の簡単な例など、Liquid フォーマットのタグおよびそれらが 3scale システム内で機能する仕組みについて説明します。

Liquid の基本を知るには、[Liquid リファレンス](#) を参照してください。

### 9.1. デベロッパーポータルでの LIQUID の使用

本セクションでは、レイアウトおよびページで Liquid マークアップ処理を有効にする方法を説明します。

#### 9.1.1. Liquid の有効化

Liquid マークアップ処理は、すべてのパーシャルとメールテンプレートに対してデフォルトで有効です。レイアウトで Liquid を有効にするには、system\_name 入力フィールドの右下にあるチェックボックスを選択します。ただし、ページで有効にする場合は、そのページの ADVANCED OPTIONS セクションに移動する必要があります。

##### ▼ ADVANCED OPTIONS

System name

Content type

Can be [HTML](#), [CSS](#), [Javascript](#) or an arbitrary MIME type.

Liquid enabled

Process Liquid tags and drops?

Handler

Do you use any markup language?

Tag list

**ADVANCED OPTIONS** セクションを展開して、Liquid enabled チェックボックスを選択します。これ以降、Liquid マークアップはすべて内部エンジンで処理され、デベロッパーポータルの組み込みエディターでも、Liquid コードが強調表示されます。

#### 9.1.2. ページ、パーシャル、およびレイアウトでの使用方法の違い

Liquid の使用法は通常、ページ、パーシャル、およびレイアウトによって若干異なります。ページ内では、Liquid は使用が 1 回限りの要素ですが、Liquid をパーシャルやレイアウトと併用すると、デベロッパーポータルで再利用できる要素になります。つまり、さまざまなページに少しずつ変更を加えて複数のレイアウトまたはパーシャルを適用するのではなく、論理 Liquid タグを追加して、ユーザーが操作するページに応じてレイアウトを変更できるということです。

```
<!-- if we are inside '/documentation' URL -->
<li class="{% if request.request_uri contains '/documentation' %}active{% endif %}"><!-- add the
active class to the menu item -->
  <a href="/documentation">Documentation</a>
</li>
```

### 9.1.3. CSS/JS との使用

Liquid マークアップは HTML で機能するだけでなく、CSS や JavaScript コードと簡単に組み合わせ、制御を強化することもできます。スタイルシートまたは JS で Liquid を有効にするには、これをページとして作成し、通常のページで有効にするのと同じ手順に従います。これにより、CSS に条件マークアップを追加することや、JavaScript でサーバー側のデータを使用することが可能です。ページのコンテンツタイプを CSS または JS として設定するのを忘れないでください。

## 9.2. メールテンプレートでの LIQUID の使用

本セクションでは、Liquid タグを使用してメールテンプレートをカスタマイズする方法を説明します。

### 9.2.1. デベロッパーポータルでの使用との相違点

前述のように、Liquid タグを使用してユーザーに送信されるメールテンプレートをカスタマイズすることもできます。前述した Liquid を記述する際の一般規則は、すべてメールテンプレートにも適用されますが、一部の例外があります。

- すべてのテンプレートで利用可能な、一般に共有される変数リストはありません。代わりに、前述の `{% debug:help %}` タグを使用してテストを行う必要があります。
- 電子メールは本質的に Web ページとは異なるため、使用に制約のあるタグや使用できないタグがあります。たとえば、電子メールには URL がないため、`{{ request.request_uri }}` は意味を持ちません。

## 9.3. トラブルシューティング

このトラブルシューティングセクションは、一般的に発生する可能性のあるエラーをデバッグして修正する場合に役立ちます。

### 9.3.1. デバッグ

正常に保存はされているが何かが意図したとおりに機能しない場合は、以下の点を確認してください。

- すべてのタグが正しく閉じられていること。
- 現在のページで利用可能な変数を参照していること。
- 配列へのアクセスを試みていないこと (たとえば、`current_account.applications` はアプリケーションの配列です)。
- 論理が正しいこと。

### 9.3.2. 典型的な誤りとその解決方法

- Liquid エラーのためにドキュメントを保存できない原因は、通常タグまたはドロップが正しく閉じられていないことです。`{% %}` タグや `{{ }}` タグがすべて正しく閉じられていること、また

if や for などの論理式が `endif` や `enfor` などで正しく終了していることを確認します。この場合は通常、エディターの上のページ最上部に、エラーを説明するメッセージと共にエラーが表示されます。

- すべてが正しく保存されているにも関わらずタグの効果が見られない場合、空の要素を参照していないか、また、コンテンツの表示に論理タグを使用していないかを確認します。すでにより複雑なタグとドロップのセットのエイリアスであるタグでの使用を除き、`{% %}` がコンテンツをレンダリングすることはありません。
- # 記号のみが表示される場合は、表示しようとしている要素が配列であることを意味します。Liquid リファレンスの階層セクションを確認してください。

### 9.3.3. サポートへの問い合わせ

問題が解決しない場合は、[Red Hat カスタマーポータル](#) から新しいケースを作成することができます。

## 第10章 LIQUID: メールテンプレート

3scale では、ご自分の組織独自のメッセージと用語を使用して、メールテンプレートをカスタマイズする機能を提供します。Liquid ドロップを活用して、各顧客に合わせてカスタマイズした情報を表示することもできます。

デベロッパーポータルで Liquid ドロップが使用されるのと同様に、各メールテンプレートには独自のコンテキストがあります。つまり、あるメールテンプレートで使用可能な Liquid ドロップが、他のメールテンプレートでも使用可能であるとは限りません。

本章では、内容ごとにグループ化されたメールテンプレートおよびそこでサポートされる Liquid ドロップのセットを使用して、どの Liquid ドロップをどこで使用できるのかについて概要を説明します。

### 10.1. アカウント管理

これらのメールテンプレートは、アカウント管理カテゴリに分類されます。

- Buyer Account confirmed
- Buyer Account approved
- Buyer account rejected

これらのテンプレートでは、以下の Liquid ドロップを使用できます。

- **user** ⇒ **User**
- **domain** ⇒ **String**
- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **support\_email** ⇒ **String**

また、以下のテンプレートも使用できます。

- Password recovery for buyer  
これらの Liquid ドロップへのアクセスがあること。
- **user** ⇒ **User**
- **provider** ⇒ **Provider**
- **url** ⇒ **url**

追加のユーザーをアカウントに招待するメールテンプレート

- Invitation  
では、以下の Liquid ドロップを使用できます。
- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **url** ⇒ **url**

## 10.2. クレジットカードに関する通知

- Credit card expired notification for provider
- Credit card expired notification for buyer

以下の Liquid ドロップを使用できます。

- **user\_account** ⇒ **Account**
- **account** ⇒ **Account**
- **provider\_account** ⇒ **Provider**
- **provider** ⇒ **Provider**

## 10.3. 制限に対するアラート

- Alert notification for provider ( $\geq 100\%$ )
- Alert notification for buyer ( $\geq 100\%$ )
- Alert notification for provider ( $< 100\%$ )
- Alert notification for buyer ( $< 100\%$ )

では、以下の Liquid ドロップを使用できます。

- **application** ⇒ **Application**
- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **service** ⇒ **Service**
- **alert** ⇒ **Alert**

## 10.4. アプリケーション

以下のメールテンプレートは、すべてアプリケーションおよびアプリケーションプランの通知に関するものです。

- Application created for provider

これらは以下の Liquid ドロップを使用できます。

- **url** ⇒ **url**

アプリケーションプランの変更リクエストに関する通知のメールテンプレート:

- Plan change request for buyer
- Plan change request for provider

これらは以下の Liquid ドロップを使用できます。



- **application** ⇒ **Application**
- **provider** ⇒ **Provider**
- **account** ⇒ **Account**
- **user** ⇒ **User**
- **plan** ⇒ **Plan**
- **credit\_card\_url** ⇒ **credit\_card\_url**

以下のメールテンプレートには、以下のような利用可能なドロップが多数含まれています。

- Application plan changed for buyer
- Application plan changed for provider
- Application trial period expired for buyer

これらは以下の Liquid ドロップを使用できます。

- **provider** ⇒ **Provider**
- **account** ⇒ **Account**
- **user** ⇒ **User**
- **plan** ⇒ **Plan**

上記のすべての Liquid ドロップに加え、アプリケーションプランに関するメッセージ

- Application suspended for buyer
- Application accepted for buyer
- Application rejected for buyer
- Application contract cancelled for provider

では、さらに以下の Liquid ドロップを使用できます。

- **application** ⇒ **Application**
- **service** ⇒ **Service**

アプリケーションキーに関する以下のメールテンプレートでは、さらに使用できる Liquid ドロップが増えます。

- Application key created for buyer
- Application key deleted for buyer
- **key** ⇒ **key**

## 10.5. 請求

### メールテンプレート

- Review invoices prior to charging for provider

では、以下の Liquid ドロップを使用できます。

- **provider** ⇒ **Provider**
- **url** ⇒ **String**

さらに、テンプレート

- Invoice charge failure for provider without retry
- Invoice upcoming charge for buyer
- Invoice charge failure for provider with retry
- Invoice charge failure for buyer without retry
- Invoice charged successfully for buyer
- Invoice charge failure for buyer with retry

では、以下の Liquid ドロップをすべて使用できます。

- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **cost** ⇒ **cost**
- **invoice\_url** ⇒ **invoice\_url**
- **payment\_url** ⇒ **payment\_url**

## 10.6. サービス

### メールテンプレート

- Service contract cancelled for provider
- Service trial period expired for buyer
- Service plan changed for provider
- Service contract suspended for buyer

では、以下の Liquid ドロップを使用できます。

- **provider** ⇒ **Provider**
- **account** ⇒ **Account**
- **user** ⇒ **User**
- **plan** ⇒ **Plan**

上記の Liquid ドロップに加え、サービステンプレート

- Service created for provider
- Service accepted for buyer
- Service rejected for buyer

では、さらに以下の Liquid ドロップを使用できます。

- **service** ⇒ **Service**
- **service\_contract** ⇒ **Contract**
- **subscription** ⇒ **Contract**

## 10.7. サインアップ

メールテンプレート

- Sign-up notification for provider
- Sign-up notification for buyer

では、以下の Liquid ドロップを使用できます。

- **user** ⇒ **User**
- **provider** ⇒ **Provider**
- **url** ⇒ **activate\_url**

## 第11章 デベロッパーポータルのレイアウトのカスタマイズ

実際のブランディングに合わせて、デベロッパーポータル全体の外観や操作感をカスタマイズできます。カスタマイズを簡単に行うための出発点として、標準の CSS スタイルシートを使用できます。レイアウトのテンプレートを作成するには、開始点として **Main layout** のコードを使用します。

このチュートリアルでは、デベロッパーポータルに独自の CSS カスタマイズを追加し、リロードして新しいスタイルの変更を反映させる方法について説明します。

### 11.1. 新規 CSS ファイルの作成

デフォルトのスタイルシート **default.css** があります。これは非常に大きく複雑であるため、これを拡張するよりも、独自のカスタム用に専用のスタイルシートを作成し、デフォルトを上書きする方が適切です。ページの作成と同じ方法で、新しいスタイルシートを作成します (高度なページ設定で適切な MIME コンテンツタイプを選択するのを忘れないでください)。

選択したレイアウトが空であることが重要です。そうでないと、ページレイアウトの HTML で CSS ルールが分かりにくくなります。

### 11.2. ページレイアウトへのスタイルシートのリンク

各レイアウトテンプレート (または共通の HEAD セクションがある場合はパーシャル) で、**bootstrap.css** へのリンクの後に、カスタム CSS へのリンクを追加します。以下に例を示します。

```
<link rel="stylesheet" href="/stylesheets/custom.css">
```

これで、独自のブランディングを楽しむことができます。

### 11.3. ページレイアウトテンプレートの定義

一般的な考え方としては、ポータルのさまざまなページスタイルごとに、個別のレイアウトを定義します。開始時に **Main layout** と呼ばれる標準レイアウトが1つあります。このレイアウトはすべてのシステム生成ページで使用されるので、デベロッパーポータルの使用に十分に慣れるまでは、このレイアウトには変更を加えないようにしてください。

通常、ポータルのホームページには、独自のスタイルを設定します。**Main layout** テンプレートが、カスタマイズの出発点となります。ページレイアウトのテンプレートを作成するには、以下を実行します。

1. **Main layout** を開き、そのコードをクリップボードにコピーします。
2. 新しいレイアウトを作成し、タイトルとシステム名を指定して、**Liquid enabled** を選択します。
3. **Main layout** コードを新しいレイアウトに貼り付けます。
4. 新規レイアウトからこの行を削除して、サイドバーメニューを削除します。

```
{% include 'submenu'%}
```

5. コードをカスタマイズしてレイアウトのテンプレートを作成します。

## 第12章 組み込みページの変更

本章では、システム生成ページの任意の要素を変更または表示/非表示にする方法について説明します。

システムによって生成される要素には、Signup、Dashboard、Account ページなど、デベロッパーポータルから変更できないものがあります。本ガイドでは、CSS および Javascript を使用してこれらのページのコンテンツをカスタマイズする方法について説明します。

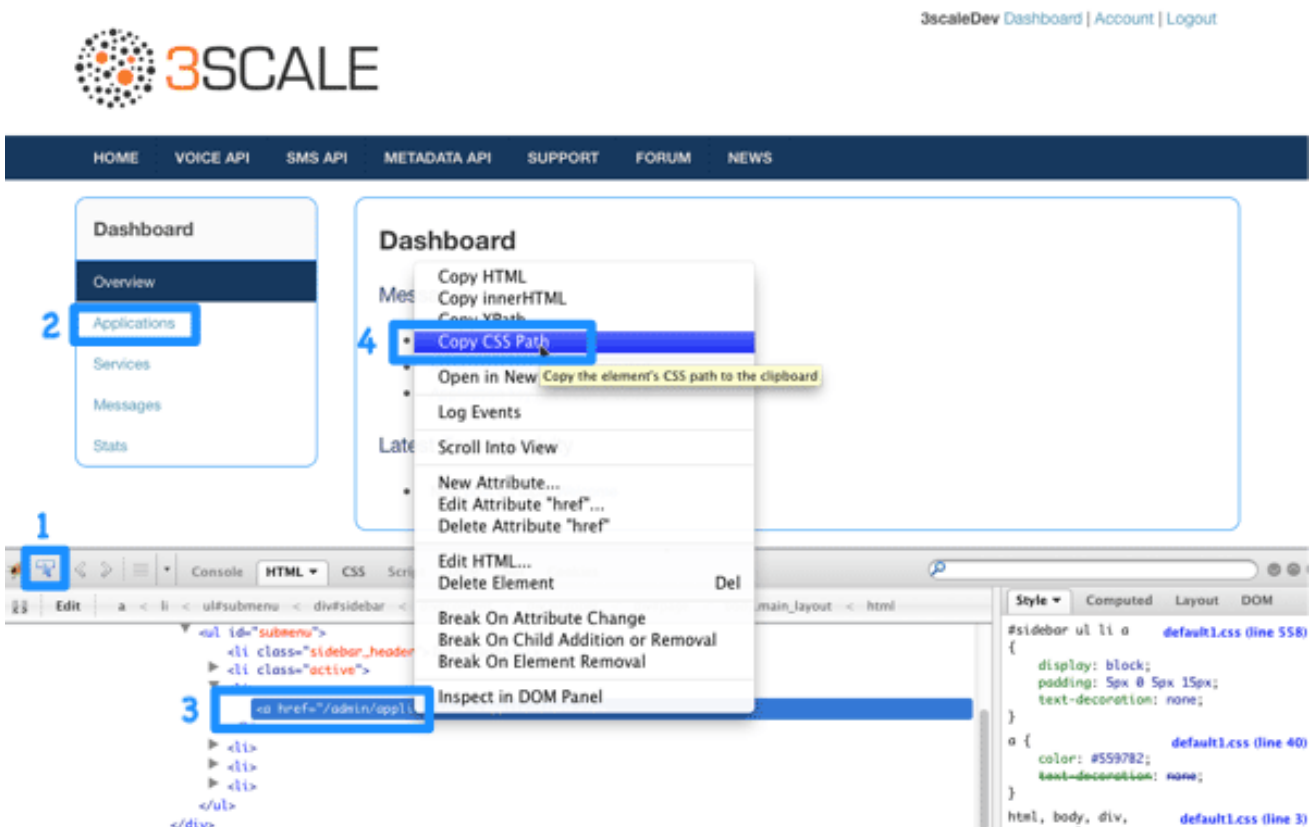
システムによって生成されたページは、アクセスおよび可視性に関するバックエンドルールに従うため、それらの URL は特定のハードコードされた値である必要があります。カスタマーポータルのレッジベース記事には、[システム生成ページとその URL のリスト](#) が記載されています。

### 注意

3scale のシステム生成ページは、変更の対象です (ただしまれです)。これらの変更により、本ガイドに従って実装するカスタマイズが機能しなくなる可能性があります。これらの変更手法を使用せずに済む場合は、使用を避けてください。設定変更を続行する前に、サービスの提供を阻害する変更を識別し、ポータルの機能を正常な状態に維持するのに必要なメンテナンス作業を実施できることを確認してください。

### 12.1. 要素の特定

何よりも重要なことは、非表示にする対象の特定です。これには、Firebug (または Chrome Developer や Opera Dragonfly など、その他の開発者ツール) を使用します。目的の要素を選択し、コンソールでその要素を右クリックして Copy CSS Path を選択します。これにより、正確な CSS パスを保存して、操作を簡単にすることができます。要素がサイドバーナビゲーションウィジェットの一部である場合は、リスト内での位置も指定する必要がある点に注意してください。これには、+セクター (たとえば、3 番目の li 要素を選択する場合: `ul + li + li + li`) または `:nth-child(n)` CSS3 疑似クラスのどちらかを使用できます。



## 12.2. 要素の変更または非表示

要素を特定したので、表示設定を変更することができます。要素のタイプに応じて、CSS 操作または jQuery スクリプトという 2 つの手法からどちらかを選択できます。CSS 操作の方が軽量で信頼性に優れますが、多くのページに存在するある種の要素に対しては適切に機能しません (たとえば、管理ポータルの Dashboard にあるサイドバーの 3 番目の要素は、Account セクションにも存在するが、値が違う)。一部のテクニカルな実装には、古いブラウザではサポートされない CSS3 の使用が必要です。次の 2 つの手順で、これらの両アプローチを説明します。

### 12.3. オプション A: CSS

たとえば、Dashboard ページから最新の要素を非表示にしてみます。最初のステップで、その CSS パスを以下のように特定しました。

```
#three-scale .dashboard_bubble
```

同じパスを持つ 2 番目のボックスなので、+セクターを使用する点に留意してください。これで、パスは以下ようになります。

```
.main_layout #three-scale .dashboard_bubble + .dashboard_bubble  
/* or */  
.main_layout #three-scale .dashboard_bubble:nth-child(1)
```

display プロパティを none に変更すると、そのボックスが非表示になります。

```
.main_layout #three-scale .dashboard_bubble:nth-child(1) {  
  display: none;  
}
```

### 12.4. オプション B: JQUERY

サイドバーメニューの要素など、テクニカルな要素を非表示にする場合は、jQuery を使用する方が適切です。これらの要素の CSS パスは Dashboard と Account セクションとで同一ですが、両方のセクションで要素を非表示にする必要はありません。そこで、CSS パスとコンテンツに基づいて要素を選択します。ここでの例は、Dashboard のサイドバーから messages セクションを非表示にする場合を想定しています。CSS パスは以下のとおりです。

```
#three-scale #submenu li a
```

コンテンツと一致させるには、`.text()` 関数を使用します。また、ドキュメントのヘッドと ready 関数内にコードを含め、すべてのコンテンツが生成された後に実行されるようにします。

The screenshot shows the 3scale Developer Portal interface. The left sidebar has a 'Layouts' section with 'main\_layout' selected. The main content area is titled 'Layout 'Main layout'' and contains a form with the following fields:

- Title: Main layout
- System name: main\_layout
- Liquid enabled Process Liquid tags and drops?

Below the form are 'Save' and 'Publish' buttons, and a status indicator showing 'Draft' and 'Published' options. A code editor displays the following HTML and JavaScript code:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <title>{{ page.title }}</title>
6 <link rel="stylesheet" href="/stylesheets/screen.css" />
7
8
9
10
11 <script>
12 $(document).ready(function() {
13 //the jQuery code for hiding elements
14 });
15 </script>
16
17
18 {% scale_essentials %}
19 </head>

```

The JavaScript code snippet is highlighted with a blue box.

作成されるコードスニペットは以下のようになります。

```

$(function() {
  $('#three-scale #submenu li a').each(function() {
    if ($(this).text() == "Messages")
      $(this).parent().css('display', 'none');
  });
});

```

これが唯一のソリューションという訳ではありません。実行可能な方法の1つを示しているだけです。属性の値に基づき Pure.CSS と CSS3 セレクターを使用しても、例と同じことができます。CSS3 セレクターの詳細な仕様については、[こちら](#)を参照してください。

## 第13章 契約条件の設定

開発者がご自分の API にサインアップするのを許可する場合、アクセス権限を付与する前に開発者に契約条件に同意してもらい、ポリシーを明確にする必要が生じます。

開発者に守ってもらう契約条件には、さまざまなバージョンが存在する可能性があります。これらは、登録プロセス中のさまざまなステップで簡単に設定できます。以下に例を示します。

1. サインアップの契約条件
2. アプリケーションの契約条件
3. サービス/サブスクリプションの契約条件 (複数のサービスがある場合のみ利用可能)

さらに、API の使用を有料にする場合は、クレジットカードポリシーを明示しなければならない場合があります。3scale では、以下のようなクレジットカードポリシーの URL を簡単に設定することができます。

1. 法律上の条件
2. プライバシー
3. 払い戻し

### 13.1. 契約条件

ワークフローのこの部分は、下記の手順に従い、管理ポータルで簡単に設定することができます。

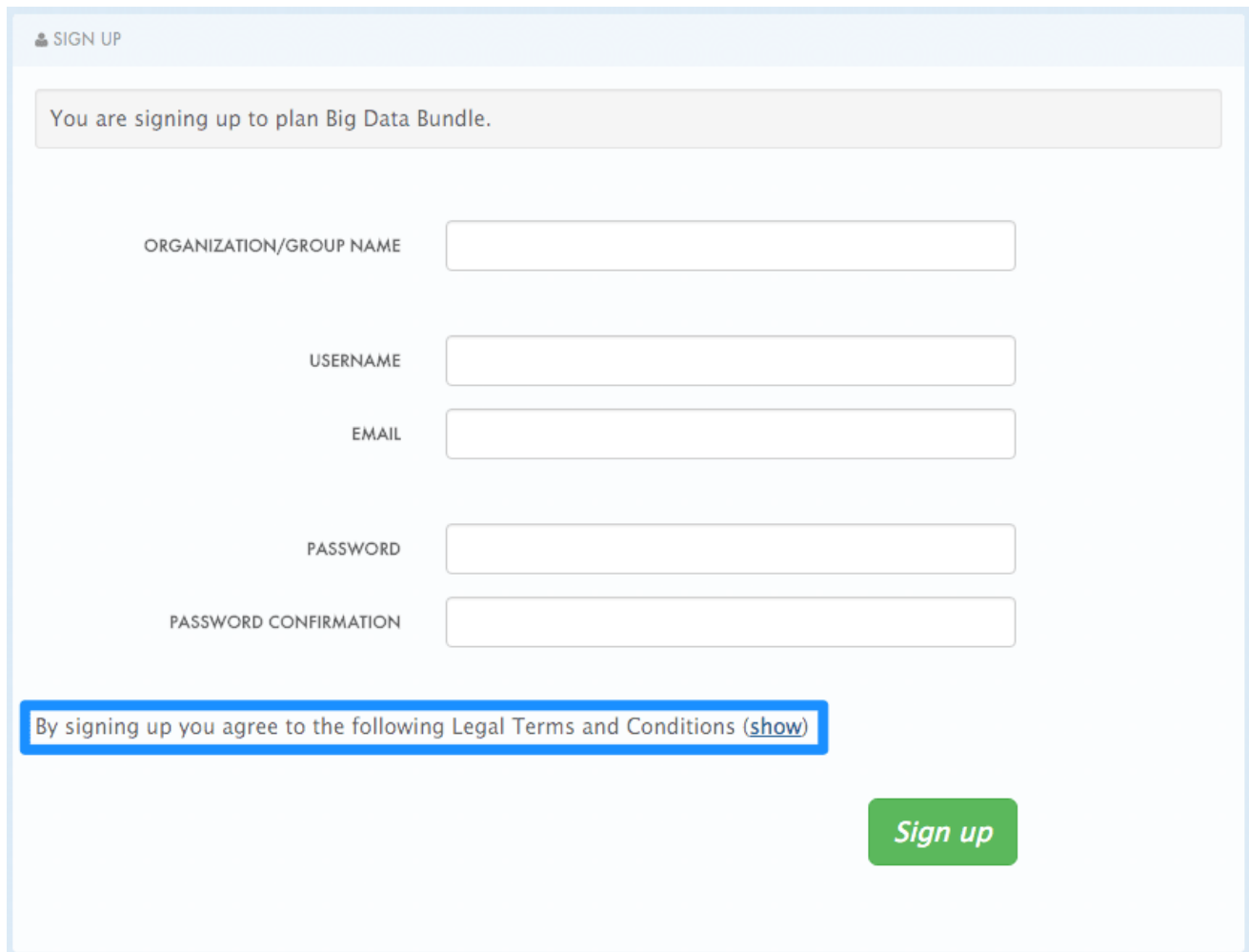
**Audience > Developer Portal > Signup**の順に移動すると、サインアップの契約条項を入力するブランクページが表示されます。HTML、JavaScript、および CSS を自由に組み合わせて使用できます。また、Insert toggling code をクリックすると、トグルコードが提示されます。このボックスで作成するコンテンツは、デベロッパーポータルの Signup ページで Sign Up ボタンのすぐ上に表示されます。

The screenshot shows the management interface for Red Hat 3scale API Management. At the top, the breadcrumb navigation is 'Audience > Developer Portal > Signup'. The main heading is 'Legal Terms for Signup'. Below the heading, there are instructions: 'When signing up, your developers have to accept these terms.' and 'Use any combination of custom HTML, Javascript and CSS to craft your legal terms. You can also insert the most common one-line warning with Show/Hide toggle using the button below.' An 'Expert Note' states: 'Legal terms are just partials included by default next to the submit button of the form. You can edit them using the CMS too. If you remove the include statement from the pages that use those snippets, these settings will no longer have any effect.' At the bottom, there is a large text input area with the number '1' inside, indicating where to enter the legal terms.



契約条件を入力したら、Update をクリックして保存します。

トグルコードを使用している場合、By signing up you agree to the following Legal Terms and Conditions の後に、指定した契約条件の表示と非表示を切り替えるリンクが表示されます。



The screenshot shows a sign-up form titled "SIGN UP". At the top, a message states "You are signing up to plan Big Data Bundle." Below this are five input fields: "ORGANIZATION/GROUP NAME", "USERNAME", "EMAIL", "PASSWORD", and "PASSWORD CONFIRMATION". A blue box highlights the text "By signing up you agree to the following Legal Terms and Conditions (show)". At the bottom right, there is a green "Sign up" button.

この契約条件は、デフォルトでは Signup ページに配置されますが、デベロッパーポータルのどこにでも含めることができるパーシャル (signup\_licence) です。Signup ページからこの契約条件を削除するには、ページから `{% include 'signup_licence' %}` の行を削除するだけです。同様に、他の場所にこれを含める場合には、スニペットによって同じパーシャルを使用し、デベロッパーポータルのどこにでも配置することができます。

また、ユーザーが新しいアプリケーションを作成する場合 (`new_application_licence` パーシャル)、または新しいサービスをサブスクライブする場合 (`service_subscription_licence` パーシャル) には、別の契約条件のセットを承諾してもらう必要が生じます。これらを設定するには、上で概説したものと同一手順に従います。

## 13.2. クレジットカードポリシー

さまざまなポリシーを規定する他の URL を定義することもできます。Audience > Billing > Credit Card Policies の順に移動し、ポリシーページの場所のパスを設定して、URL の設定を行います。

これらのリンクを機能させるためには、デベロッパーポータルに新しいページを作成する必要があります。

完了したら、URL の Liquid ドロップを使用してこれらを参照することができます。以下に例を示します。

```
<a href="{{ urls.credit_card_terms }}">Legal Terms</a>
<a href="{{ urls.credit_card_privacy }}">Privacy</a>
<a href="{{ urls.credit_card_refunds }}">Refunds</a>
```

これで完了です。