



Red Hat 3scale API Management 2.5

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 2.5 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

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 © 2022 | You need to change the HOLDER entity in the en-US/Creating_the_Developer_Portal.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 API Management 2.5 のデベロッパーポータルについて説明します。

目次

第1章 デベロッパーポータル	4
1.1. デベロッパーポータルの概要	4
1.2. CONTENT	4
1.3. LAYOUTS および PARTIALS	6
1.4. PORTLETS	7
1.4.1. 注記	7
1.5. REDIRECTS および CHANGES	8
第2章 カスタムのサインアップフォームフィールド	9
第3章 サインアップフローの設定	15
3.1. すべての承認ステップの削除	15
3.2. すべてのデフォルトプランの有効化	16
3.3. ワークフローのテスト	17
第4章 マルチサービスへのサインアップ	18
4.1. 前提条件	18
4.2. はじめに	18
4.3. マルチサービスへのサインアップ	18
4.3.1. サービスに関する情報の取得	18
4.3.2. サインアップカラムの設定	19
4.3.3. サブスクリプションの設定	19
4.3.4. スタイリング	20
第5章 デベロッパーポータルの認証	21
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	22
5.2. GITHUB による認証の有効化および無効化	22
5.3. AUTHO による認証の有効化および無効化	24
5.3.1. 注記	24
5.4. RED HAT SINGLE SIGN-ON による認証の有効化および無効化	24
5.4.1. 操作を始める前に	24
5.4.2. RH SSO の設定	25
5.4.3. 3scale の設定	28
第6章 デベロッパーポータル用の RED HAT SINGLE SIGN ON	30
6.1. 3SCALE プラットフォームでのユーザーの作成	30
6.2. ログインリンクのリクエスト	30
6.3. 自動ログインによるユーザーのリダイレクト	30
第7章 コンテンツの制限	32
7.1. ページの制限	32
7.2. コンテンツブロックの制限	33
7.3. 追加フィールドの設定の自動化	33
7.4. ユーザーログインの要求	34
第8章 メールテンプレート	35
8.1. メールテンプレートのカスタマイズ	35
8.1.1. メール設定の前に行うワークフローの定義	35
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	35
8.1.3. カスタムテンプレートの編集および保存	35
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	36
8.2. 補足情報	36

第9章 LIQUID: デベロッパーポータル	37
9.1. デベロッパーポータルでの LIQUID の使用	37
9.1.1. Liquid の有効化	37
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	37
9.1.3. CSS/JS との使用	38
9.2. メールテンプレートでの LIQUID の使用	38
9.2.1. デベロッパーポータルでの使用との相違点	38
9.3. トラブルシューティング	38
9.3.1. デバッグ	38
9.3.2. 典型的な誤りとその解決方法	38
9.3.3. サポートへの連絡	39
第10章 LIQUID: メールテンプレート	40
10.1. アカウント管理	40
10.2. クレジットカードに関する通知	41
10.3. 制限に対するアラート	41
10.4. アプリケーション	41
10.5. 請求	42
10.6. サービス	43
10.7. サインアップ	44
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	45
11.1. 新規 CSS ファイルの作成	45
11.2. ページレイアウトへのスタイルシートのリンク	45
第12章 組み込みページの変更	46
12.1. 要素の特定	46
12.2. 要素の変更または非表示	46
12.3. オプション A: CSS	47
12.4. オプション B: JQUERY	47
第13章 WEBHOOK	49
13.1. WEBHOOK の概要	49
13.2. WEBHOOK のフォーマット	49
13.3. トラブルシューティング	50
第14章 契約条件の設定	51
14.1. 契約条件	51
14.2. クレジットカードポリシー	52
第15章 要約: デベロッパーポータルのビギナーからエキスパートに	54
15.1. 目標	54
15.2. 前提条件	54
15.3. デベロッパーポータルの設定	54
15.3.1. ポータルのコンセプトのプランニング	54
15.3.2. 編集環境の設定	54
15.3.3. ページレイアウトテンプレートの定義	55
15.3.4. ページ階層の作成	55
15.3.5. ページヘッダーの編集	57
15.3.6. イメージおよび他のアセットの追加	57
15.3.7. ブランディングを使用したカスタマイズ	58
15.3.8. 実稼働環境への移行	58

第1章 デベロッパーポータル

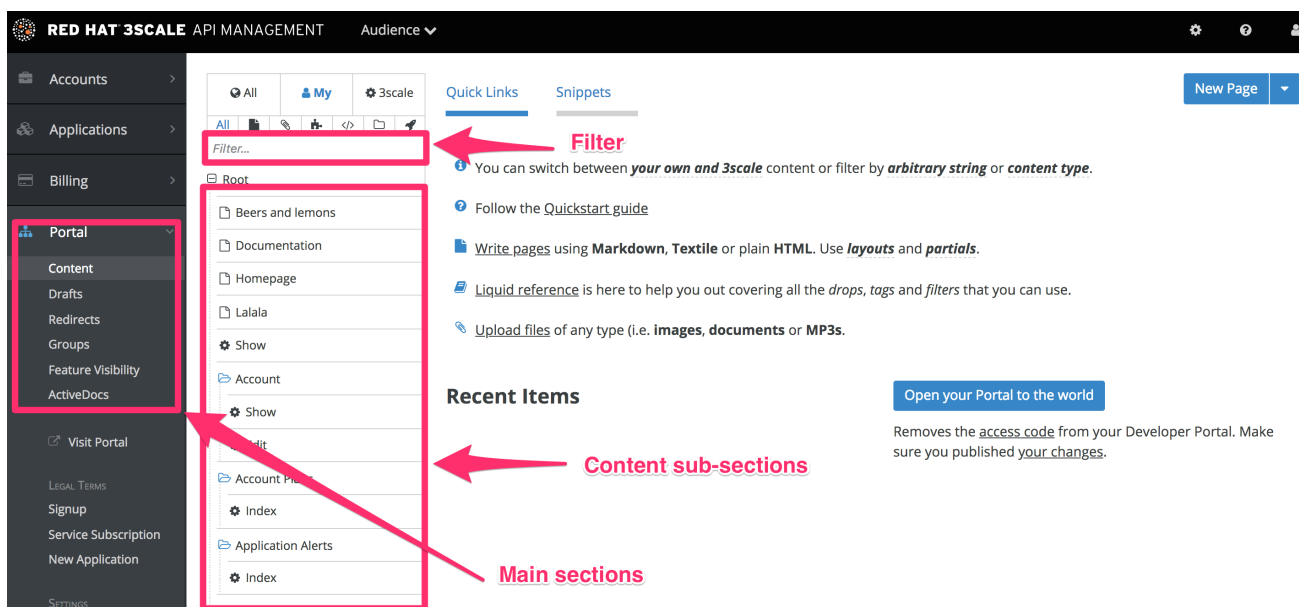
本セクションを完了すると、デベロッパーポータルの構造、使用方法、および機能を理解することができます。

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。ポータルのすべての要素を完全に制御できるので、開発者に API の使用方法を容易に理解してもらうことができます。API デベロッパーポータルが適切に作成されていれば、開発者はコンセプトをごく短時間のうちに実際のアプリケーションに具体化することができます。

1.1. デベロッパーポータルの概要

デベロッパーポータルの設定要素は以下のとおりです。

- 左側のメニュー: Content、Drafts、Redirects、Groups、Feature Visibility、ActiveDocs、Legal Terms、Settings、および Liquid のドキュメントにアクセスできます。
- メインエリア: 上記セクションの詳細が表示されます。



1.2. CONTENT

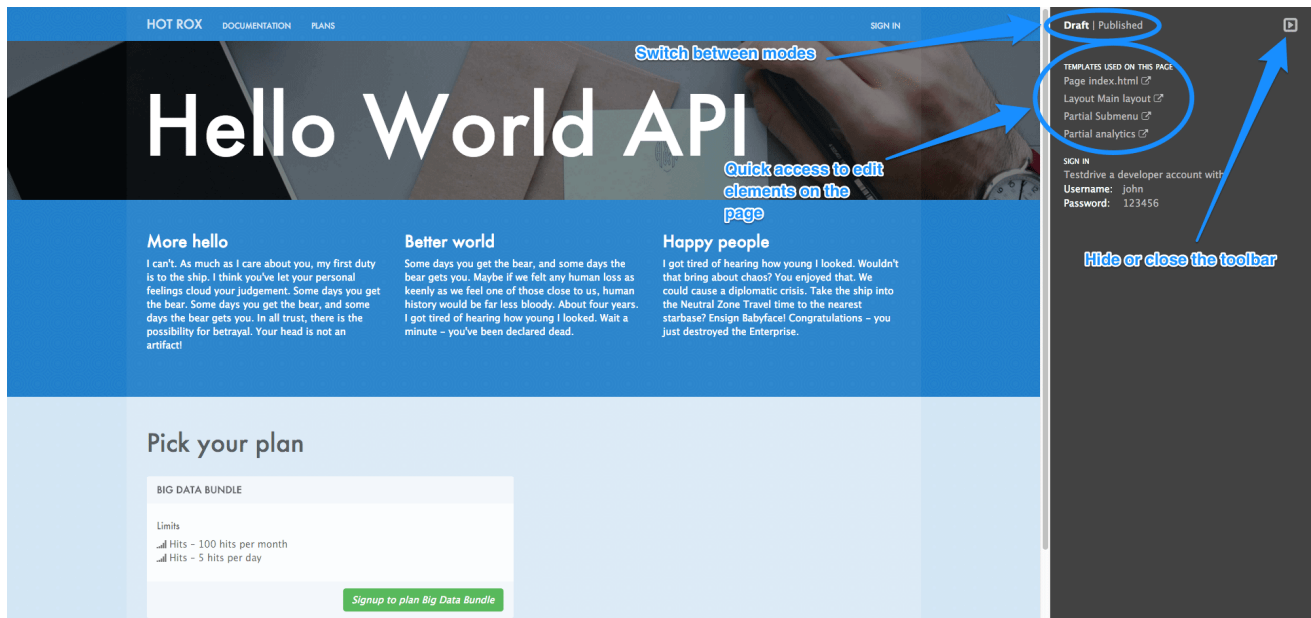
このセクションは、デベロッパーポータルシステムビューで最も重要な部分です。Content セクションにはサイト構造と階層が表示され、またページ内のコンテンツを編集することができます。つまり、サイト構造、ページ、およびそこに保存されたその他のアセットを管理できます。ポータルの階層は、ディレクトリーツリーの形式で表示されます。

上記の図は、Content セクション内のあるページのサンプルビューを示しています。図に示すように、サイトのパス階層を維持したまま、すべてのファイル（ページ、イメージ、スタイルシート、JavaScript など）が表示されます。前述のように、セクションは機能的にディレクトリーと同じです。

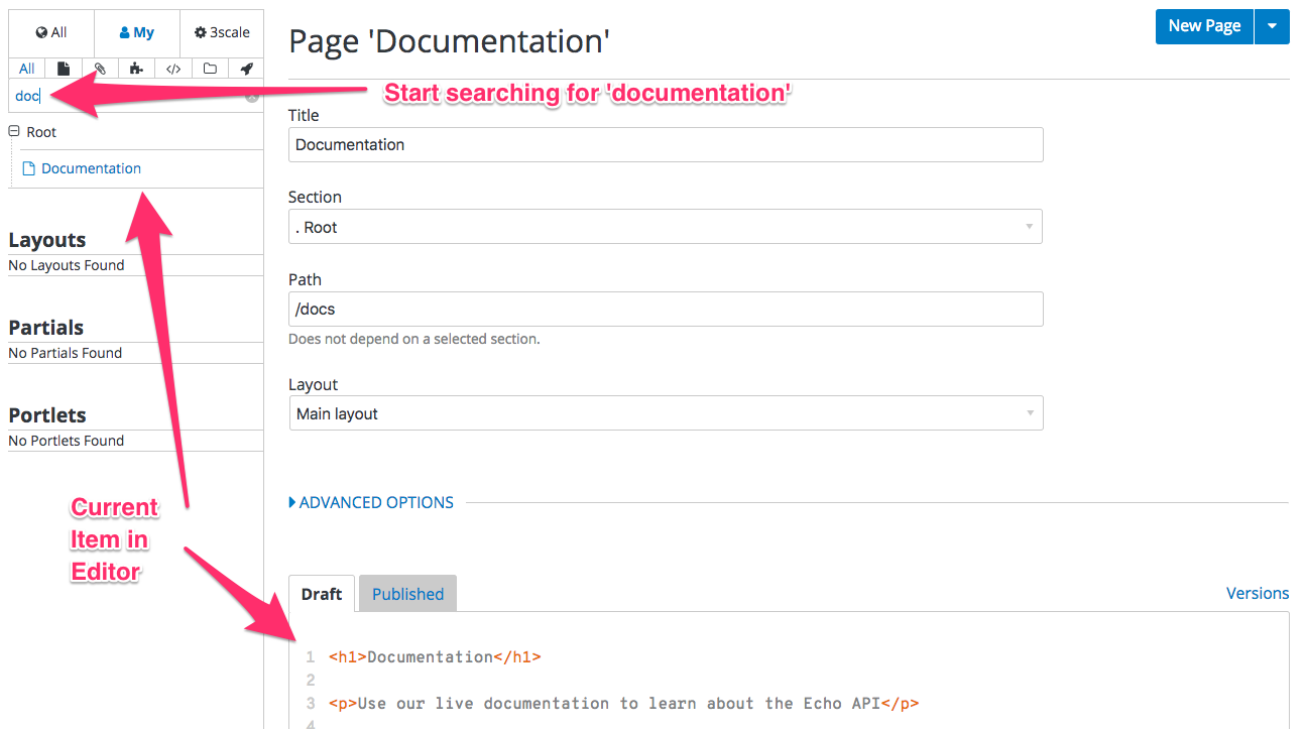
右側には、ページ編集ビューが表示されます。ここにはページ名（標準ページか組み込みページかどうか）も示され、コンテンツに新しい要素（ページ、レイアウト、パーシャル、セクション、ファイル、またはポートレット）を追加するためのボタンが表示されます。その下では、ページで使用するレイアウトを選択したり、Liquid タグの機能を切り替えたりすることができます。それに続く部分はテキストエディターで、コードの強調表示、表作成、行番号など多くの機能に対応しています。タブボタンの Draft と Published を使用すると、編集したドキュメントのドラフトバージョンと公開バージョンが切り替わります。それに続くのは、ドキュメントのバージョンを一覧表示するアイコン、およびポップアップ編集ウィンドウを開くアイコンです。

ページのコンテンツを編集するには、目的のレイアウトを選択し、コンテンツタイプや URL パスなどのいくつかの追加オプションを設定してから、HTML、Markdown、または Textile 形式でコードを入力するだけです。

このビューのもう1つの重要な機能は、Preview ボタンです。ページの公開バージョンまたはドラフトバージョンをプレビュー表示するかどうかを選択できます。ボタンをクリックするとデベロッパーポータルモードにリダイレクトされ、レンダリングされたページの公開（またはドラフト）バージョンと共に、ダークグレーの縦長のバーが表示されます（右側）。このバーには、デベロッパーポータルのページ、レイアウト、およびパーシャルの編集ビューへのリンクが含まれています。ここでは、ドラフトと公開ビューを切り替えることができます。



フィルター機能もあり、検索フィールドとして機能するだけでなく、表示される要素をスタイルシート、JavaScript またはその他の指定タイプのみで制限することができます。



1.3. LAYOUTS および PARTIALS

Layouts および Partials セクションでは、テンプレートとページの再利用可能な部分を管理します。これらの機能は Content セクションの機能に類似しています。

Layouts セクションは、ページで使用されるテンプレートの定義で設定されます。レイアウトはページの主要構造で、このテンプレートのコンテンツが、そのレイアウトを使用するすべてのページでレンダリングされます。ページのパーシャル、ポートレット、および実際のコンテンツがテンプレートの内部に存在します。

パーシャルとはコードの再利用可能な部分で、さまざまなページのさまざまな場所で繰り返し使用されます。たとえば、すべてのレイアウトで同じフッターを使用したり、レイアウトの異なるいくつかのページで同じサイドバーを使用したりすることができます。レイアウト、パーシャル、またはメールテ

ンプレートやポートレットにパーシャルを含めるには、`{% include "partial_name" %}` と入力します。Liquid タグの詳細な情報は、[こちら](#) を参照してください。

ポータルの他の部分と同様、レイアウトとパーシャルにもドラフトと公開の状態があり、完全なバージョン履歴を確認できます。

1. レイアウトテンプレート用のテキストエディター
2. ドラフトを保存する、現在のバージョンを公開する、最後の公開バージョンに戻すことができます。
3. テキストエディターをドラフトバージョンと公開バージョンで切り替える、バージョン履歴を一覧表示する、ポップアップエディターを起動することができます。

1.4. PORTLETS

Content の最後のサブセクションは Portlets です。難易度の高いコーディングを必要とすることなく、高度な機能を利用できます。デベロッパーポータルでは、以下の3つの異なるポートレットを利用することができます。

- 外部 RSS フィード: 指定されたソースから RSS フィードを取得します
- 目次: 指定のセクションにページのリンクのリストを生成します
- 最新のフォーラム投稿: 最新のフォーラム投稿のリストを生成します

必要なポートレットを作成する際には、タイトル、システム名、および外部 RSS フィードポートレットの URL フィードなど、必要なデータを設定ページで入力する必要があります。

1.4.1. 注記

エディターには、カスタム Liquid タグを使用する標準のポートレットコードが事前に入力されます。生成された構造の編集は可能ですが、注意して行ってください。不明な点があれば、**Liquid Reference** を参照してください。

1.5. REDIRECTS および CHANGES

デベロッパーポータルの最後の要素は、Redirects セクションと Changes セクションです。これらは、Content セクションよりもかなりシンプルですが、やはり重要なセクションであり、いくつかのカスタム機能を利用することができます。

Redirects は、あるポータル URL から別のポータル URL へのリダイレクトを設定するのに役立ちます。これは、たとえば古いページを非推奨にするが、すべてのリンクを変更したくない場合に便利です。Redirects は組み込みのデベロッパーポータルページには使用できません。**自分で作成したページ**にしか使用することはできません。

最後の Changes セクションも重要です。ここには新しく編集され公開されていないすべてのページの一覧が含まれ、個別に公開するか、すべてを1度に公開するかを選択できます。

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。


The screenshot shows the 'Fields Definitions' page in the Red Hat 3Scale API Management portal. The page is divided into three sections: Account, User, and Application. Each section contains a list of fields with their names, descriptions, and status (Required, Edit, Create).

Account				
org_name	"Organization/Group Name"	Required	Edit	Create

User				
username	"Username"	Required	Edit	Create
email	"Email"	Required	Edit	Create

Application				
name	"Name"	Required	Edit	Create
description	"Description"	Required	Edit	Create

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 \(show\)](#)

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

FIELD DETAILS

Name

The low level system name.

Label

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

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 API 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。たとえば、[3scale API Docs](#) の 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。

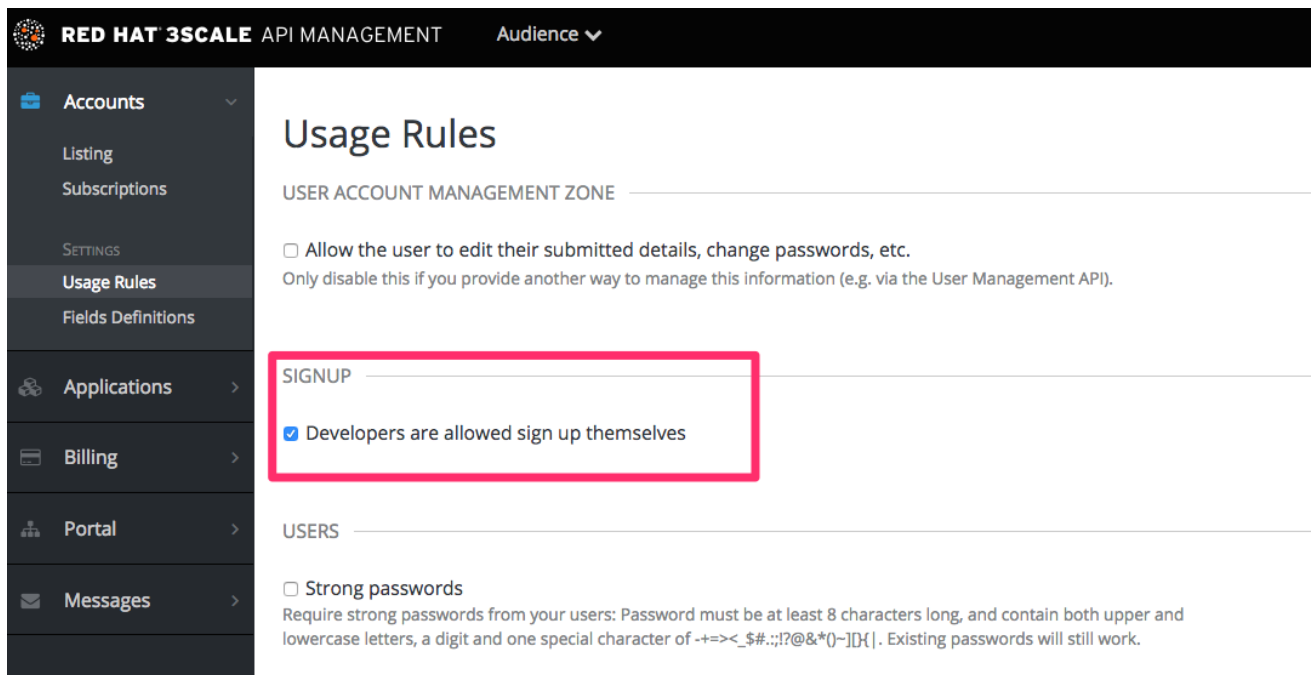
サインアップワークフローは、デベロッパーポータルを通じて提供する開発者体験の最も重要な部分です。このプロセスは、完全自動のセルフサービスから、逆に誰が何にアクセスできるかを完全に管理するものまで、さまざまな粒度に対応しています。

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 3scale API Management interface. The left sidebar contains a navigation menu with 'Accounts' expanded, showing 'Listing' and 'Subscriptions'. Under 'SETTINGS', 'Usage Rules' is selected. The main content area is titled 'Usage Rules' and has a sub-section 'USER ACCOUNT MANAGEMENT ZONE'. It contains two options: 'Allow the user to edit their submitted details, change passwords, etc.' (unchecked) and 'Developers are allowed sign up themselves' (checked). The 'SIGNUP' section is highlighted with a red box. Below this is the 'USERS' section with the 'Strong passwords' option (unchecked).

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

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 Reference](#) 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。

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

```

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 title is 'Single Sign-on Integrations'. Below the title is a table with two columns: 'Integration' and 'State'. The table contains three rows of integration entries:

Integration	State
GitHub	Hidden
★ Auth0	Hidden
Red Hat Single Sign-On	Hidden

The left sidebar contains a navigation menu with items such as Content, Drafts, Redirects, Groups, Feature Visibility, ActiveDocs, Visit Portal, LEGAL TERMS, Signup, Service Subscription, New Application, SETTINGS, Domains & Access, Spam Protection, SSO Integrations (highlighted), and Forum Settings.

GitHub をクリックして設定画面にアクセスします。

The screenshot shows the GitHub integration settings page. At the top is the GitHub logo and an 'Edit' button. Below that are three settings:

- Published:** A checked checkbox.
- Branding:** Set to '3scale branded'.
- Authentication Flow:** Set to '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。

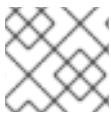
3scale 管理ポータルで **Audience > Developer Portal > SSO Integrations**の順に移動し、**Auth0** をクリックします。

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 is titled 'New Auth0 Authentication Provider' and contains the following fields and options:

- Client ID**: A text input field.
- Client Secret**: A text input field.
- Site**: A text input field with an example: 'Ex: https://XXXXX.auth0.com'.
- Published**: A checkbox labeled 'Published' with the subtext 'Display on Developer Portal'.
- Create Auth0**: A blue button at the bottom right.

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. 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、レルムを作成します。
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 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/*)。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。

The screenshot shows the 'Mappers' configuration page for the client '3scale-dev-portal'. The left sidebar contains navigation options: Master, Configure, Realm Settings, Clients (selected), Client Templates, Roles, Identity Providers, User Federation, Authentication, and Manage. The main content area shows a table of mappers with columns for Name, Category, Type, and Actions. The table contains five entries:

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。

The screenshot shows the 'Add Builtin Protocol Mapper' page. The left sidebar is the same as in the previous screenshot. The main content area shows a table of builtin mappers with columns for Name, Category, Type, and Add. The table contains five entries:

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"/>

Below the table is a blue button labeled 'Add selected'.

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	openid-connect
Name	org_name
Consent Required	<input type="checkbox"/> OFF
Mapper Type	User Attribute
User Attribute	org_name
Token Claim Name	org_name
Claim JSON Type	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 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。

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

- b. 名前を付けて、識別できるようにします。
- c. **Mapper Type** で **Attribute Importer** を選択します。
- d. **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。
- e. **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に、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. 3scale の設定

Red Hat Single Sign-On 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 Red Hat Single Sign-On 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、**Red Hat Single Sign-On** をクリックします。(備考: これはエンタープライズ専用の機能なので、この機能を有効にするようアカウントマネージャーに依頼しなければならない場合があります。)

以下の設定画面で、前の手順で設定した Red Hat Single Sign-On クライアントの詳細情報を追加する必要があります。

- **Client:** Red Hat Single Sign-On でのクライアントの名前
- **Client Secret:** Red Hat Single Sign-On でのクライアントシークレット
- **Realm:** レルム名および Red Hat Single Sign-On への URL アドレス

これらを入力したら、**Published** チェックボックスを選択し、**Create RH-SSO** をクリックしてデベロッパーポータルで利用できるようにします。

RED HAT 3SCALE API MANAGEMENT Audience ▾

Content
Drafts
Redirects
Groups
Feature Visibility
ActiveDocs

Visit Portal

LEGAL TERMS
Signup
Service Subscription
New Application

SETTINGS

Domains & Access
Spam Protection
SSO Integrations
Forum Settings

Docs
Liquid Reference

New Red Hat Single Sign-On Authentication Provider

Client ID

Client Secret

Realm

e.g.: `https://rh-ssso.example.com/auth/realms/demo`

Do not verify SSL certificate

Published
Display on Developer Portal

Create Red Hat Single Sign-On

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 コンシューマーは、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注意してください。このためページには、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 3scale 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、ブラウザで `&` 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

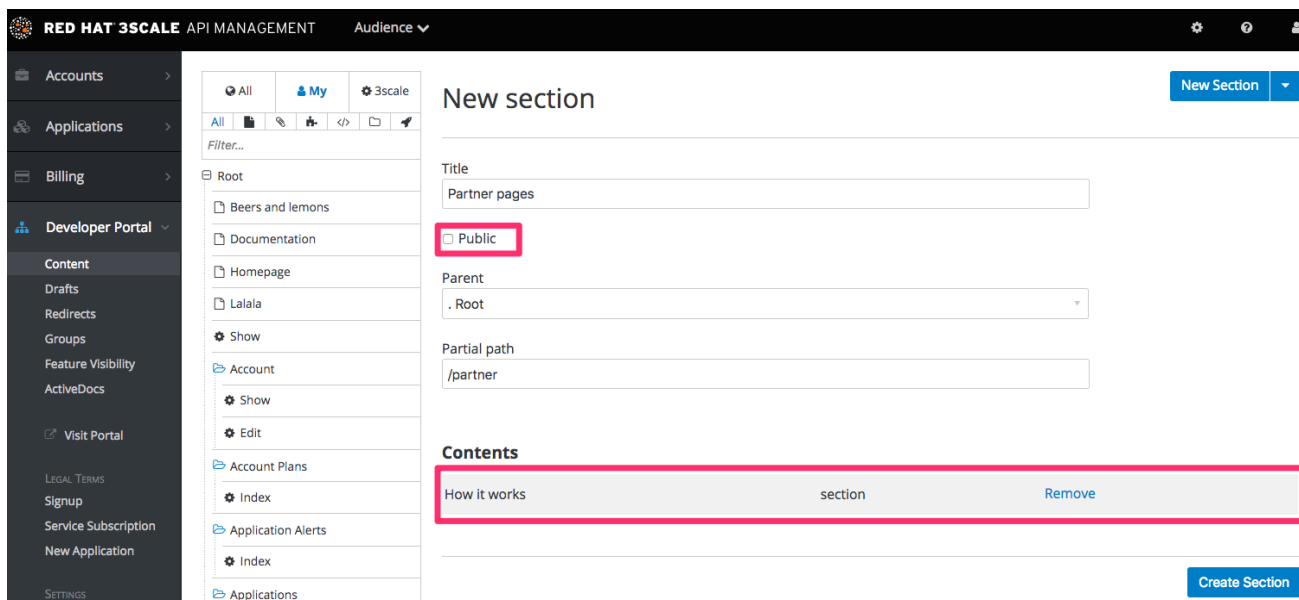
本章では、デベロッパーポータルのコンテンツを一部のユーザーにしか見えないようにする方法を説明します。

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。

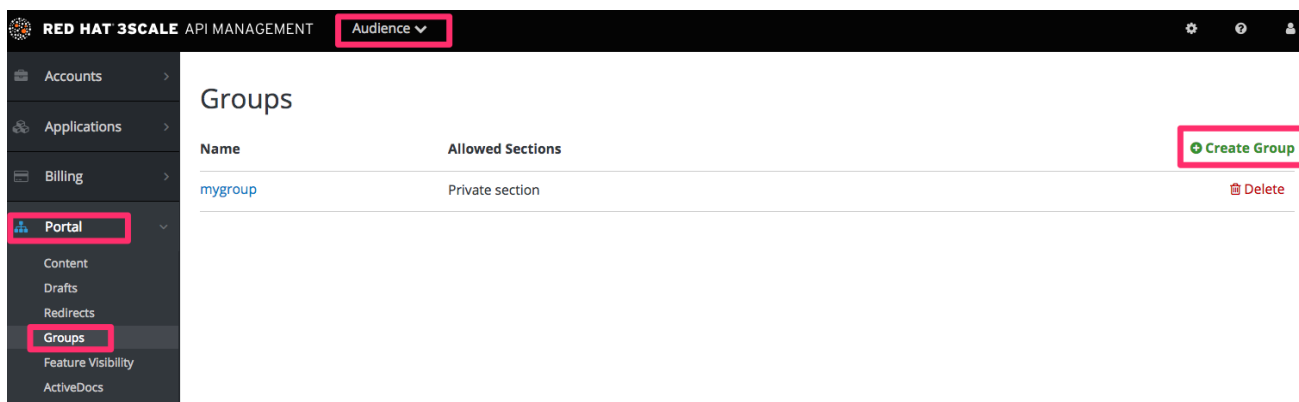
7.1. ページの制限

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。

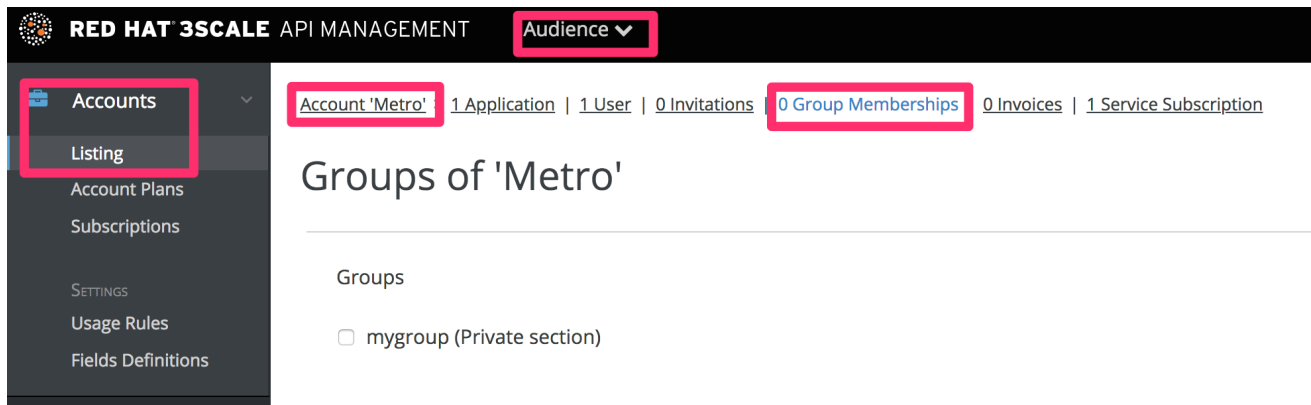
アクセスを制限するすべてのページまたはページのグループについて、CMS で新しいセクションを作成します。public ステータスフィールドのチェックボックスの選択を解除します。次に、このセクション内に必要なすべてのページをドラッグアンドドロップします。



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。

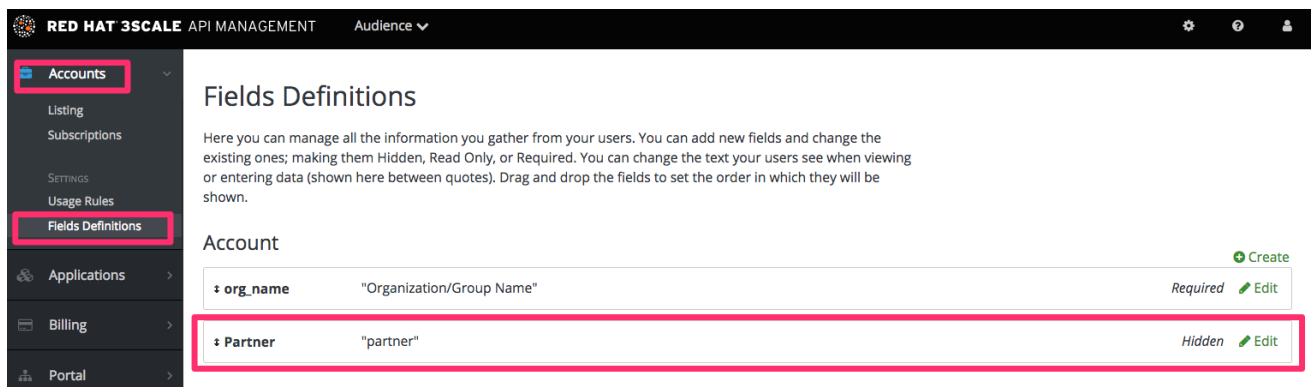


これで、このセクションへのアクセス権限をユーザーの一部に付与するときには、このグループに割り当てただけで済みます。そのためには、該当するアカウントの詳細ページから Group Permissions に移動します。移動したら、許可するセクションのチェックボックスを選択します。



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。



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

状況の変化に応じて、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すると、このプロセスを簡素化できます。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。開発者の新しいプラン (Webhook リクエストから送信されるメッセージで把握できる) 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

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するだけです。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

text/html

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

通常、ページ (ポータルで1回しか使用されない要素) とパーシャル/レイアウト (ポータルで再利用可能な要素) とでは、Liquid の使用方法が若干異なります。つまり、あまり違いのない複数のレイアウトやパーシャルをさまざまなページに適用する代わりに、論理 Liquid タグを追加し、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

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。

```
<!--samples-->
```

9.3. トラブルシューティング

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 `endfor` 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。)

- # 記号しか表示されない場合は、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 Reference で Liquid の階層に関するセクションを確認してください。

9.3.3. サポートへの連絡

それでも問題が解決しない場合は、[Red Hat カスタマーポータル](#) から新しいサポート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

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しなければならない場合があります。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。

CMS 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

以下のメールテンプレートが本セクションのカテゴリに分類されます。

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

これらのメールテンプレートでは、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
では、以下の Liquid ドロップを使用できます。
- **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 (>= 100%)
- Alert notification for buyer (>= 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、利用可能なドロップが多数含まれます。

- 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。

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。

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非表示にする方法について説明します。

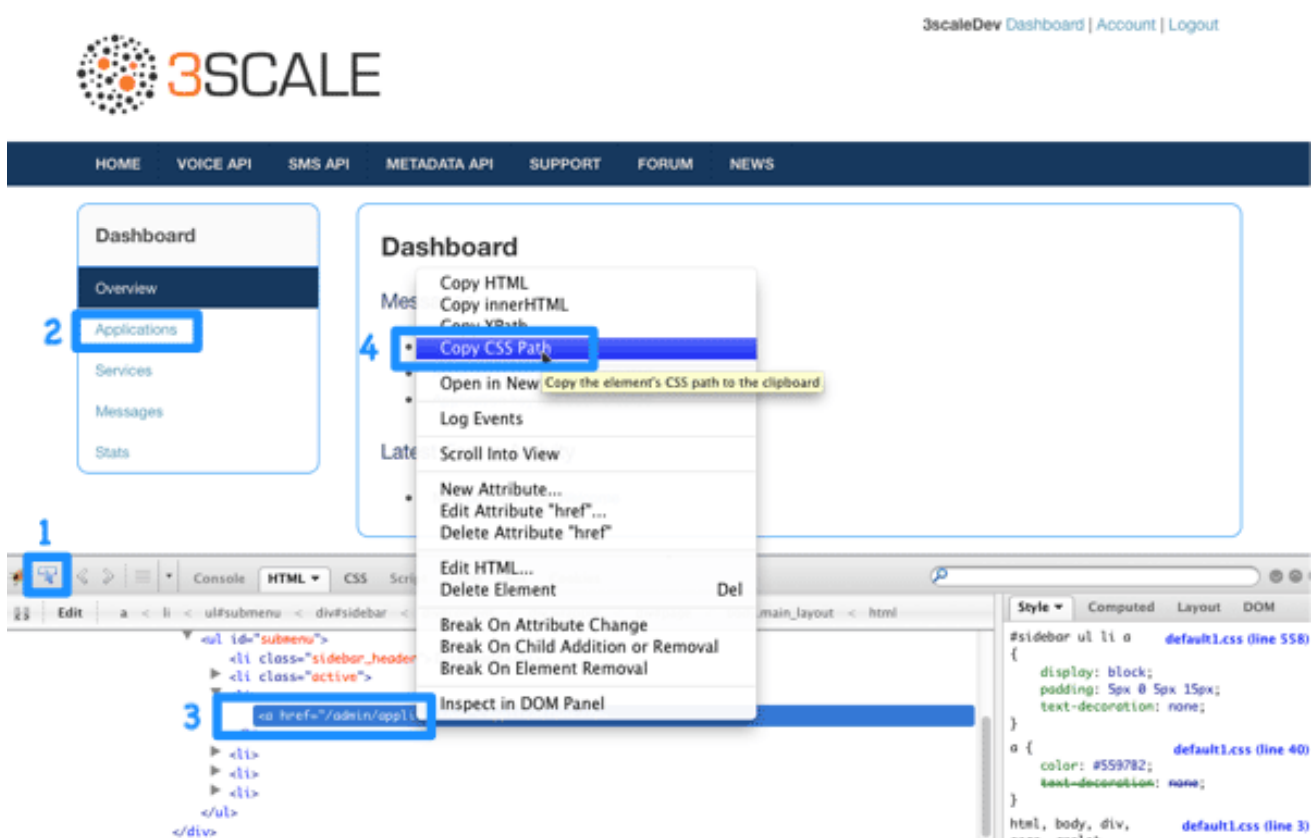
Signup ページ、Dashboard ページ、および Account ページなど、システムによって生成される要素にはデベロッパーポータルからは変更できないものもあります。本章では、CSS および JavaScript を使用して、これらのページ上のコンテンツをカスタマイズする方法を説明します。

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 疑似クラスのうち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 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、最新のフォーラム投稿ボックス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、+セレ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。

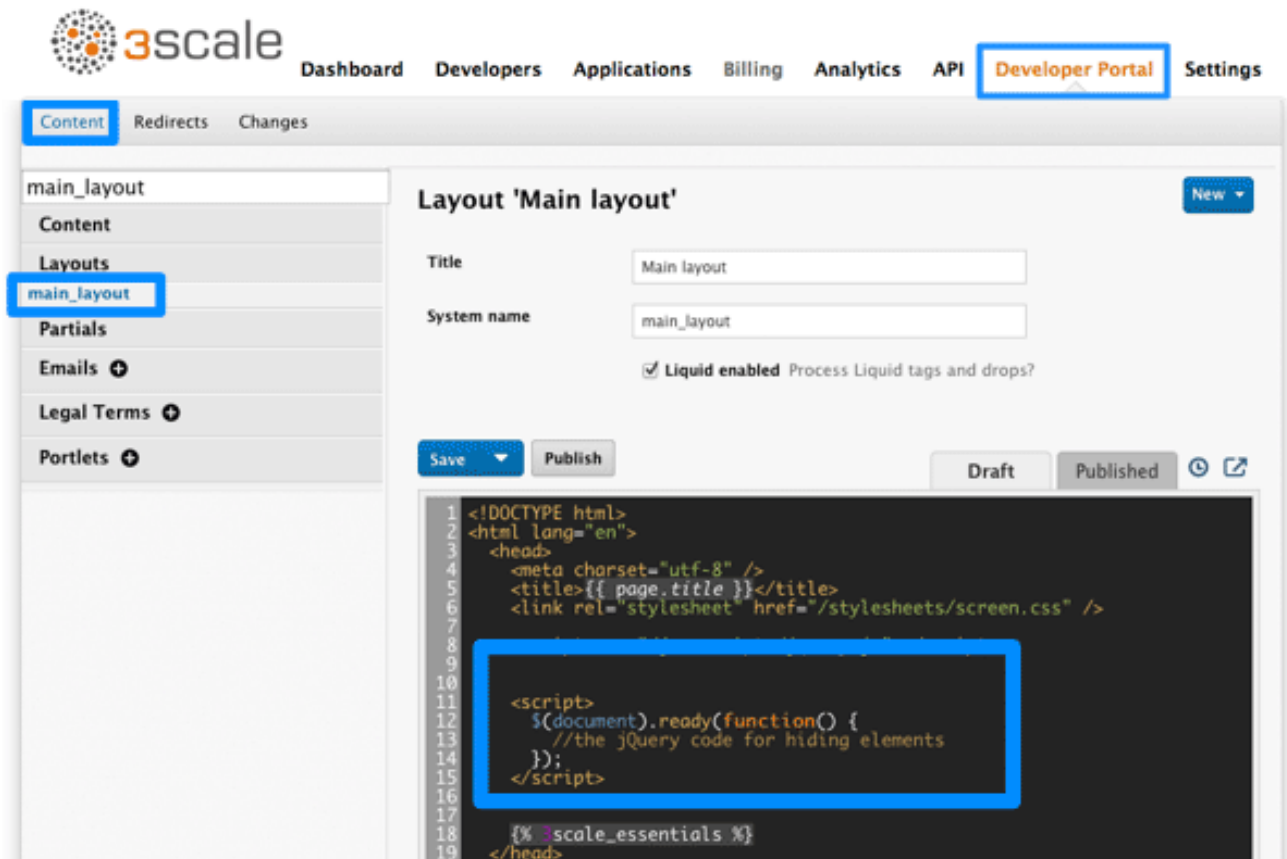
```
.main_layout #three-scale .dashboard_bubble:nth-child(1) {  
  display: none;  
}
```

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。



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。

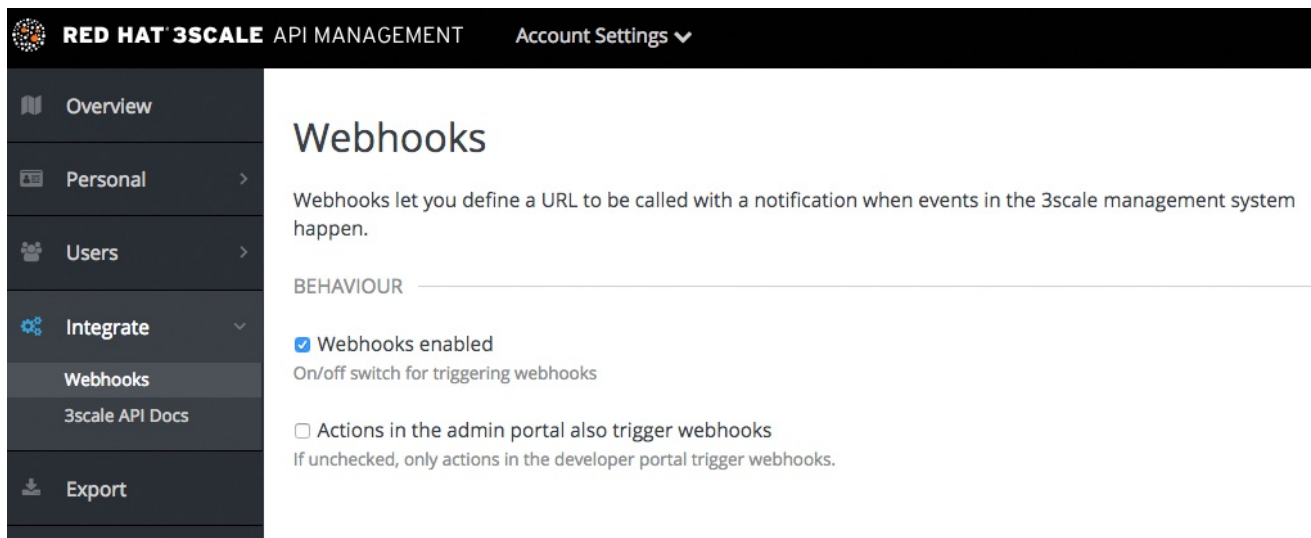
第13章 WEBHOOK

本章では、**デベロッパーポータル** で Webhook を設定し、操作を行う方法について説明します。

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統合できます。3scale システムで指定のイベントが発生すると、Webhook メッセージによりアプリケーションに通知が届き、新規アカウントのサインアップからのデータなどを使用して、CRM システムに反映させることができます。

13.1. WEBHOOK の概要

Webhook は、イベントによってトリガーされるカスタムの HTTP コールバックです。3scale システムでは、考え得るすべてのイベントが **Account Settings** (右上隅の歯車アイコン) の **Integrate > Webhooks** に表示されます。



これらのイベントのいずれかが発生すると、3scale システムは、Webhook セクションに設定された URI に対して HTTP (または HTTPS) リクエストを行います。API プロバイダー側では、リスナーを設定してイベント追跡などの目的の動作を呼び出すことができます。

スクリーンショットの2つのチェックボックスは、Webhook を有効にするためのものと (Webhooks are スイッチ)、管理ポータルのアクションでも Webhook がトリガーされるかどうかの設定です。デフォルトの動作では、デベロッパーポータル内のアクションでしか Webhook はトリガーされません。つまり、すべてのイベントでトリガーされるわけではないことを念頭に置いておいてください。

13.2. WEBHOOK のフォーマット

Webhook のフォーマットは常に同じです。以下の構造の XML ドキュメントでエンドポイントにポストします。

```
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <type>application</type>
  <action>updated</action>
  <object>
    THE APPLICATION OBJECT AS WOULD BE RETURNED BY A GET ON THE ACCOUNT
    MANAGEMENT
```

```
API
</object>
</event>
```

<type> は、application、account など、イベントの主体を表します。<action> は、updated、created、deleted など、何が行われたかを表します。最後の <object> は、Account Management API によって返される、同じフォーマットの XML オブジェクト自体です。これを確認するには、インタラクティブな ActiveDocs (管理ポータルの右上隅にある疑問符 (?) アイコン → 3scale API Docs セクションから利用可能) を使用できます。

Webhook が 3scale によって実行されたことを保証する必要がある場合は、HTTPS Webhook URL を公開し、3scale の Webhook 宣言にカスタムパラメーターを追加します。たとえば、<https://your-webhook-endpoint?someSecretParameterName=someSecretParameterValue> となります。パラメーター名と値を指定します。続いて、Webhook エンドポイント内で、このパラメーター値があることを確認します。

13.3. トラブルシューティング

Webhook を試す場合、または問題のトラブルシューティングを行う場合、Webhook の結果を表示できる RequestBin という素晴らしい (しかも無料の) サービス (<http://requestb.in/>) があります。

リッスンしているエンドポイントに障害が発生した場合、失敗した配信を回復できます。エンドポイントがコード 200 を返す場合に、3scale は Webhook が配信されたとみなします。そうでない場合は、60 秒間隔で 5 回リトライします。障害からの復旧後に、または定期的にチェックを実行し、必要に応じてキューをクリーンアップする必要があります。以下の手法についての詳細は、[ActiveDocs](#) を参照してください。

- 配信に失敗した Webhook の一覧
- 配信に失敗した Webhook の削除

第14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

14.1. 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 3scale API Management interface. At the top, there is a navigation bar with 'RED HAT 3SCALE API MANAGEMENT' and a dropdown menu for 'Audience'. On the left, a sidebar menu is visible with 'Portal' selected, and 'LEGAL TERMS' and 'Signup' highlighted with red boxes. The main content area is titled 'Legal Terms for Signup' and contains the following text:

When signing up, your developers have to accept these terms.

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.

Expert Note: 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.

Below the text is a large empty text area for editing the legal terms, with a small '1' at the top left.

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。

👤 SIGN UP

You are signing up to plan Big Data Bundle.

ORGANIZATION/GROUP NAME

USERNAME

EMAIL

PASSWORD

PASSWORD CONFIRMATION

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

[Sign up](#)

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。

14.2. 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。

RED HAT 3SCALE API MANAGEMENT Audience ▾

Accounts >

Applications >

Billing ▾

Earnings by Month

Invoices

SETTINGS

Charging & Gateway

Credit Card Policies

Portal >

Messages >

Credit Card Policies

POLICY URLS FOR CREDIT CARD DETAILS

Path to Legal Terms page
/termsofservice

Path to Privacy page
/privacypolicy

Path to Refund page
/refundpolicy

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。

RED HAT 3SCALE API MANAGEMENT Audience ▾

Accounts >

Applications >

Billing >

Developer Portal ▾

Content

Drafts

Redirects

Groups

Feature Visibility

ActiveDocs

Visit Portal

LEGAL TERMS

Signup

Service Subscription

New Application

Legal Terms for Signup

When signing up, your developers have to accept these terms.

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.

Expert Note: 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.

1

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。

第15章 要約: デベロッパーポータルのビギナーからエキスパートに

API 開発のベストプラクティスの調査からは、適切な設定のデベロッパーポータルと優れたドキュメントが、確実に API を利用してもらうための鍵であることが分かっています。デベロッパーポータルは、単なるドキュメントのソースではありません。開発者とのやり取りを管理するためのメインハブでもあり、また開発者はデベロッパーポータルを通じて API キーにセキュアにアクセスすることができます。

15.1. 目標

本チュートリアルでは、デベロッパーポータルを稼動状態にして API をプロモートし、開発者がサインアップしてアカウントを作成し、API キーにアクセスするのを許可する方法について説明します。

15.2. 前提条件

ポータルと互いに依存関係にある、設定すべきその他の項目があります。これらについては、デベロッパーポータルの作成とは別に対応することができます。

- **アプリケーションプランの設定**: これらのプランにより、発行する API キーに対するアクセス権限が設定されます。
- **開発者サインアップフローの設定**: これらのフローは、セルフサービス、承認を要するもの、または招待のみ (サインアップ無効) のいずれかです。
- **ポータルのカスタムドメインの設定** (オプション): この処理には 1-2 週間のリードタイムが発生する可能性があり、通常、カスタムの発信メールアドレスを作成すると同時に設定されません。

15.3. デベロッパーポータルの設定

15.3.1. ポータルのコンセプトのプランニング

3scale デベロッパーポータルを開設する前に、ポータルのコンセプトを綿密にプランニングすると良いでしょう。しっかり整理されているほど、この手順を効率的に完了させることができます。

プランニングの上で最も重要な要素は、以下のとおりです。

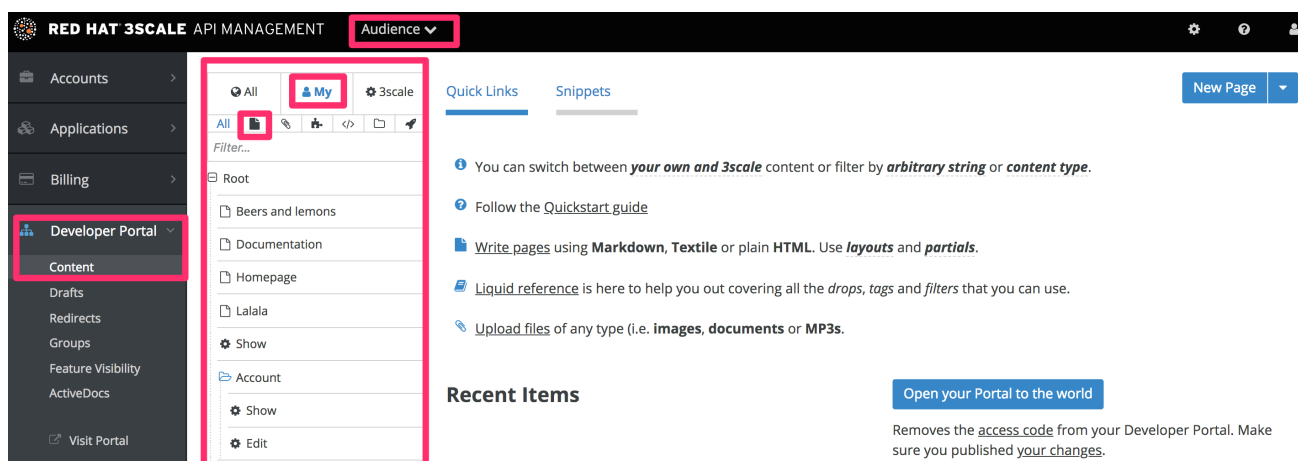
- **サイトマップ**: ポータル構造の骨格
- **トップメニューバー**: すべてのページで繰り返し表示されるナビゲーション
- **サイドメニューバー**: 各セクション内の個々のページへのアクセス用
- **ページレイアウトのガイドライン**: ポータルの外観と操作感に一貫性を持たせる

15.3.2. 編集環境の設定

編集環境に最適なセットアップは、以下のとおりです。

- 管理者クレデンシャルでログインする `yourcompany-admin.3scale.net/p/admin/cms` を示すタブ。ポータルのデベロッパーポータルにアクセスすることができます。
- ポータルの公開ビューである `yourcompany.3scale.net` をポイントする別のタブ (Site リンクでこれにアクセスする場合、アクセスコードを心配する必要はありません)。

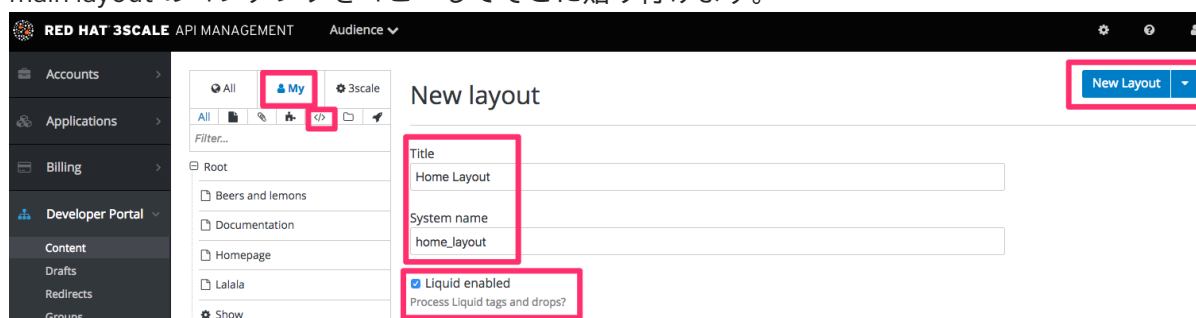
管理パネルでは、左側のサイドバーにデベロッパーポータルの要素がすべて表示されます。



15.3.3. 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。設定開始時には、main layout という1つの標準レイアウトがあります。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、何も変更すべきではありません。通常、ご自分のポータルのホームページには固有のスタイルが必要になります。

1. main layout テンプレートが、カスタマイズの出発点となります。新しいレイアウトを作成し、main layout のコンテンツをコピーしてそこに貼り付けます。



2. home layout から以下の行を削除して、サイドバーメニューを削除します。

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

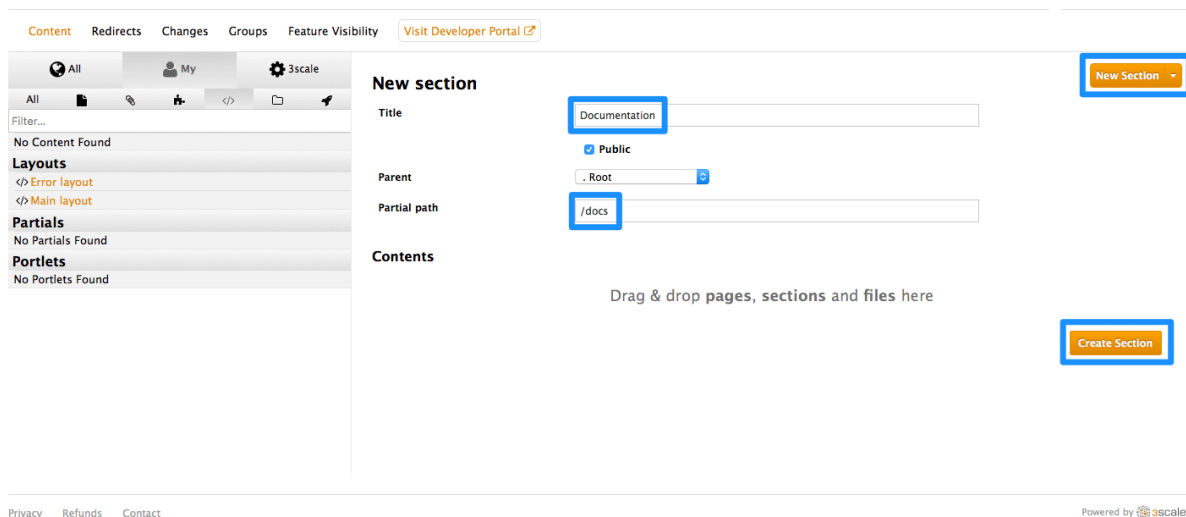


Content copied from the "Main Layout" template.

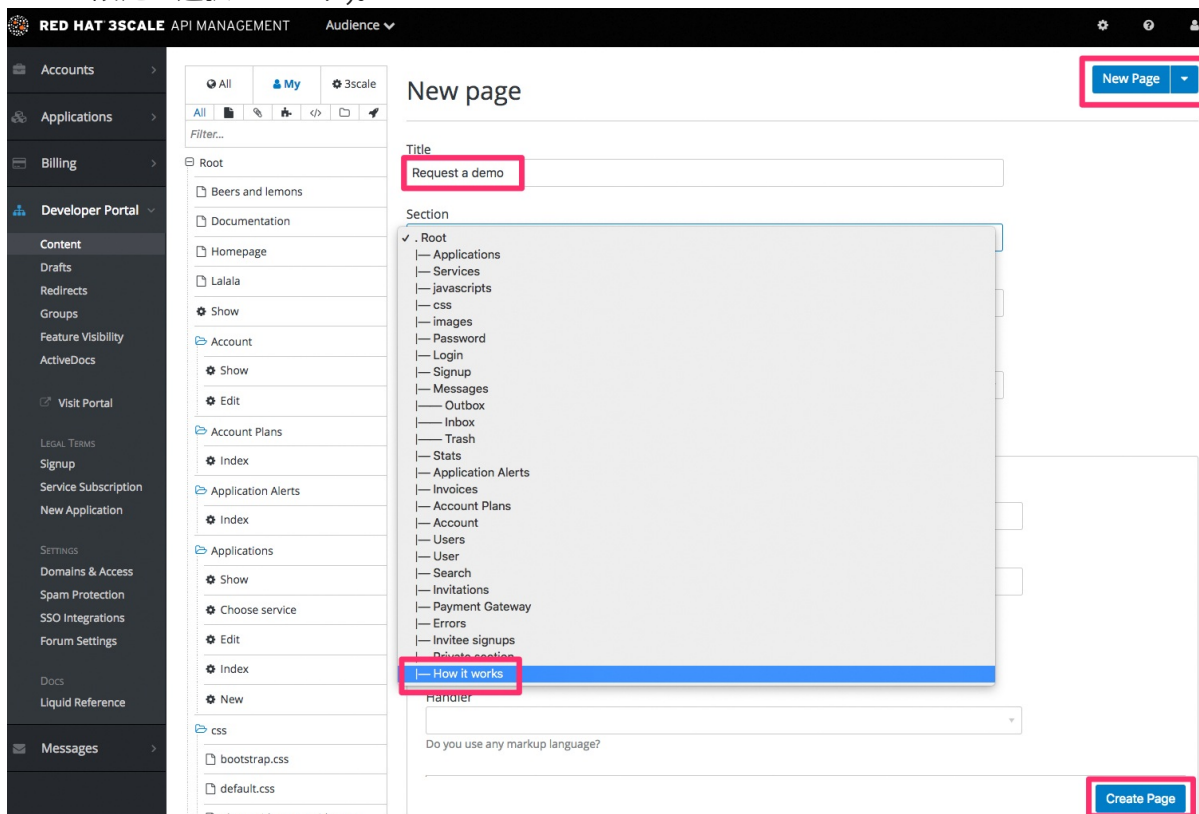
```
34 <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar-1">
35 <span class="sr-only">Toggle navigation</span>
36 <span class="icon-bar"></span>
37 <span class="icon-bar"></span>
38 <span class="icon-bar"></span>
39 </button>
40 <a class="navbar-brand" href="/" >{{ provider.name }}</a>
41 </div>
42 {% include "submenu"%}
43 </div>
44 </nav>
```

15.3.4. ページ階層の作成

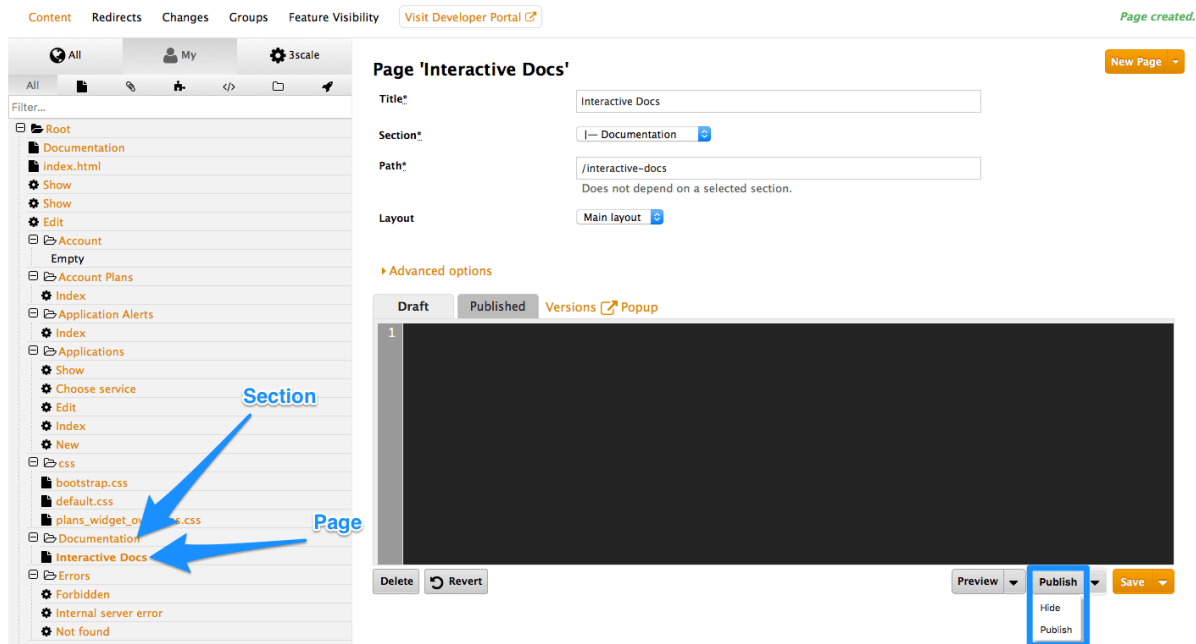
1. サイトマップのルートレベルから始め、各トップメニュー項目に対して新しいセクションを追加します (右側の new ボタンを展開してセクションを追加します)。タイトル、親セクション、およびパスを割り当てます。



2. セクションの追加と同様に、ページを追加します。URL パスの構造に一貫性を持たせるため、目的のセクションを選択します。次に、ページで使用するレイアウトを選択します。ページコンテンツが完成したら、Create Page をクリックします。多くのコンテンツを作成する場合は、Textile や Markdown などのマークアップ言語の使用が望ましい可能性があります (高度なページ設定で選択できます)。



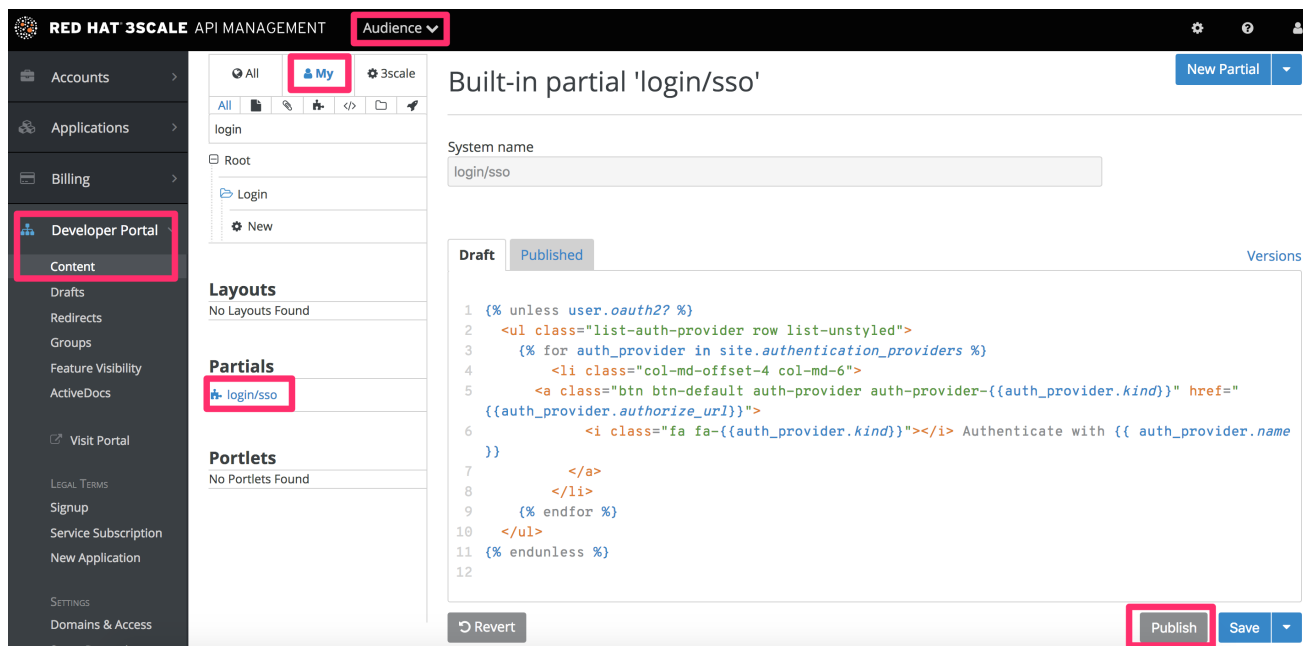
3. ドラフトのプレビューを表示し、満足のできる公開バージョンができるまで、ページコンテンツを改良します。



4. セクションのすべてのページに対して手順を繰り返します。
5. 次に、サイトのすべてのセクションに対して手順を繰り返します。

15.3.5. ページヘッダーの編集

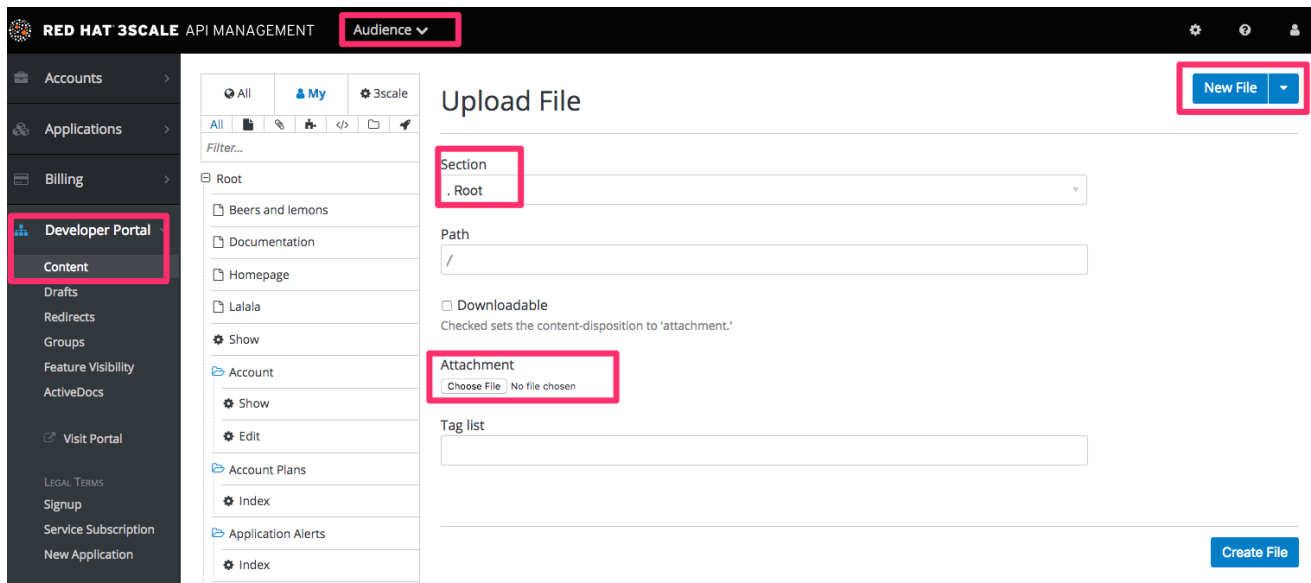
ヘッダーやフッターなどの繰り返し表示されるページ要素は、すべてポータルのデベロッパーポータルの Partials というセクションで定義されます。レイアウトが1つしかない場合、あるいは極めて少ない場合は、レイアウトコード内にヘッダーとフッターを含めることで、この手順を省略することができます。ただし、レイアウトでこれらの要素をカスタマイズするのを忘れないでください。たとえば、デフォルトの menu パーシャルを、実際のサイトマップに対応するように編集する必要があります。



15.3.6. イメージおよび他のアセットの追加

イメージまたは他のファイルについては、まずファイルをコンテンツライブラリーに読み込み、次にリンクをテキストコンテンツに挿入します。

1. ファイルを選択するために New File を選択し、これをサイト内のどこに保存するかを指定します。
2. イメージの URL をコピーします。
3. これで、HTML または <a> タグを追加して、イメージの URL に貼り付けることができます。



15.3.7. ブランディングを使用したカスタマイズ

デフォルトの default.css というスタイルシートがありますが、これは非常に大きく複雑です。これを拡張するよりも、カスタム設定で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することができます。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。次に、レイアウトテンプレートで、default.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" />
```

15.3.8. 実稼働環境への移行

最後のタスクは、ポータルサイト全体を調べ、すべてのワークフローを確認することです。各ページまたは Changes セクションのすべてのページを公開できます。すべて問題がなければ、すべてのページが公開されていることを最終確認します。

これで、ポータルのアクセスコードを削除する準備ができました。

RED HAT 3SCALE API MANAGEMENT Audience

Domains & Access

ACCESS CONTROL

Developer Portal Access Code

<no access code>

The access code hides your site from the world, but allows you to share access to a select few users during setup. Add text to the field to enable the screen. **Delete the text to open the site to the world.**

DOMAINS

Developer Portal Site

xxxxxxx

You can change the domain of your Developer Portal to your own domain, for instance https://developer.example.com. To do so, **add a route on OpenShift** and then indicate the new domain here.

Outgoing Email

xxxxxxx

You can change the domain of the email address received by your users, for instance api@example.com

Update Account

Developer Portal

Domains & Access

お疲れさまでした。これでデベロッパーポータルが公開され、開発者コミュニティ構築を支援する準備が整いました。