



Red Hat 3scale API Management 2.14

创建开发者门户

良好的开发者门户是确保 API 使用的关键。快速创建您的内容。

良好的开发者门户是确保 API 使用的关键。快速创建您的内容。

法律通告

Copyright © 2024 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[®] 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.

摘要

本指南记录了红帽 3scale API 管理 2.14 上的开发人员门户。

目录

对红帽文档提供反馈	4
第 1 章 为 3SCALE 管理的 API 创建开发者门户概述	5
1.1. 准备 3SCALE API 管理编辑环境以创建开发人员门户	5
1.2. 修改 3SCALE API 管理原生开发人员门户	7
1.3. 修改原生开发人员门户的其他 3SCALE API 管理功能描述	9
1.4. 3SCALE API 管理布局和部分重复利用开发人员门户内容	11
1.5. 在开发者门户中为 3SCALE 管理的 API 添加镜像和其他资产	11
1.6. 为 3SCALE 管理的 API 自定义原生开发人员门户的注意事项	12
1.7. 对于 3SCALE 管理的 API 允许访问开发人员门户的要求	12
1.8. 为 3SCALE 管理的 API 自定义开发人员门户的可选步骤	14
1.9. 更新早于 2.8 的 3SCALE API 管理版本中使用的开发者门户	15
第 2 章 自定义注册表单字段	18
第 3 章 配置注册流程	24
3.1. 删除所有批准步骤	24
3.2. 启用所有可能的默认计划	24
3.3. 测试工作流	25
第 4 章 多服务注册	27
4.1. 先决条件	27
4.2. 简介	27
4.3. 多服务注册	27
第 5 章 开发者门户身份验证	30
5.1. 启用和禁用用户名/电子邮件和密码	30
5.2. 通过 GITHUB 启用和禁用身份验证	31
5.3. 通过 AUTH0 启用和禁用身份验证	32
5.4. 通过 RED HAT SINGLE SIGN-ON 启用和禁用身份验证	33
第 6 章 RED HAT SINGLE SIGN ON FOR DEVELOPER PORTAL	38
6.1. 在 3SCALE API 管理平台中创建用户	38
6.2. 请求登录链接	38
6.3. 使用自动登录重定向用户	38
第 7 章 限制内容	40
7.1. 受限页面	40
7.2. 内容的受限块	41
7.3. 自动配置额外字段	41
7.4. 需要用户登录	42
第 8 章 电子邮件模板	43
8.1. 自定义电子邮件模板	43
8.2. 更多信息	44
第 9 章 LIQUIDS : 开发人员门户	45
9.1. 在 DEVELOPER PORTAL 中使用 LIQUIDS	45
9.2. 在电子邮件模板中使用 LIQUIDS	46
9.3. 故障排除	46
第 10 章 LIQUIDS : 电子邮件模板	47
10.1. 帐户管理	47
10.2. 信用卡通知	47

10.3. 限制警报	48
10.4. 应用程序	48
10.5. INVOICING	49
10.6. 服务	50
10.7. 注册	51
第 11 章 自定义开发人员门户布局	52
11.1. 创建新的 CSS 文件	52
11.2. 将风格表链接到您的页面布局	52
11.3. 定义页面布局模板	52
第 12 章 更改内置页面	53
12.1. 确定元素	53
12.2. 修改或隐藏元素	53
12.3. 选项 A : CSS	54
12.4. 选项 B: JQUERY	54
第 13 章 设置条款和条件	56
13.1. 条款和条件	56
13.2. 信用卡策略	57
第 14 章 导出和导入开发人员门户	59

对红帽文档提供反馈

我们感谢您对我们文档的反馈。

要改进，创建一个 JIRA 问题并描述您推荐的更改。提供尽可能多的详细信息，以便我们快速解决您的请求。

前提条件

- 您有红帽客户门户网站帐户。此帐户可让您登录到 Red Hat Jira Software 实例。如果您没有帐户，系统会提示您创建一个帐户。

流程

1. 单击以下链接：[创建问题](#)。
2. 在 **Summary** 文本框中输入问题的简短描述。
3. 在 **Description** 文本框中提供以下信息：
 - 找到此问题的页面的 URL。
 - 有关此问题的详细描述。
您可以将信息保留在任何其他字段中的默认值。
4. 点 **Create** 将 JIRA 问题提交到文档团队。

感谢您花时间来提供反馈。

第 1 章 为 3SCALE 管理的 API 创建开发者门户概述

您的 3scale 开发人员门户是应用程序编程接口(API)用户使用的网站：

- 注册以访问 3scale 管理的上游 API
- 有关如何使用上游 API 的文档

3scale 提供了一个示例开发者门户，它包括了大多数 API 供应商要在其开发者门户中实施的功能。这个原生开发人员门户使用示例 Echo API 来演示典型的开发人员门户的结构。在浏览原生开发人员门户后，您需要做相应的修改信息，以便您了解如何创建自己的开发人员门户。

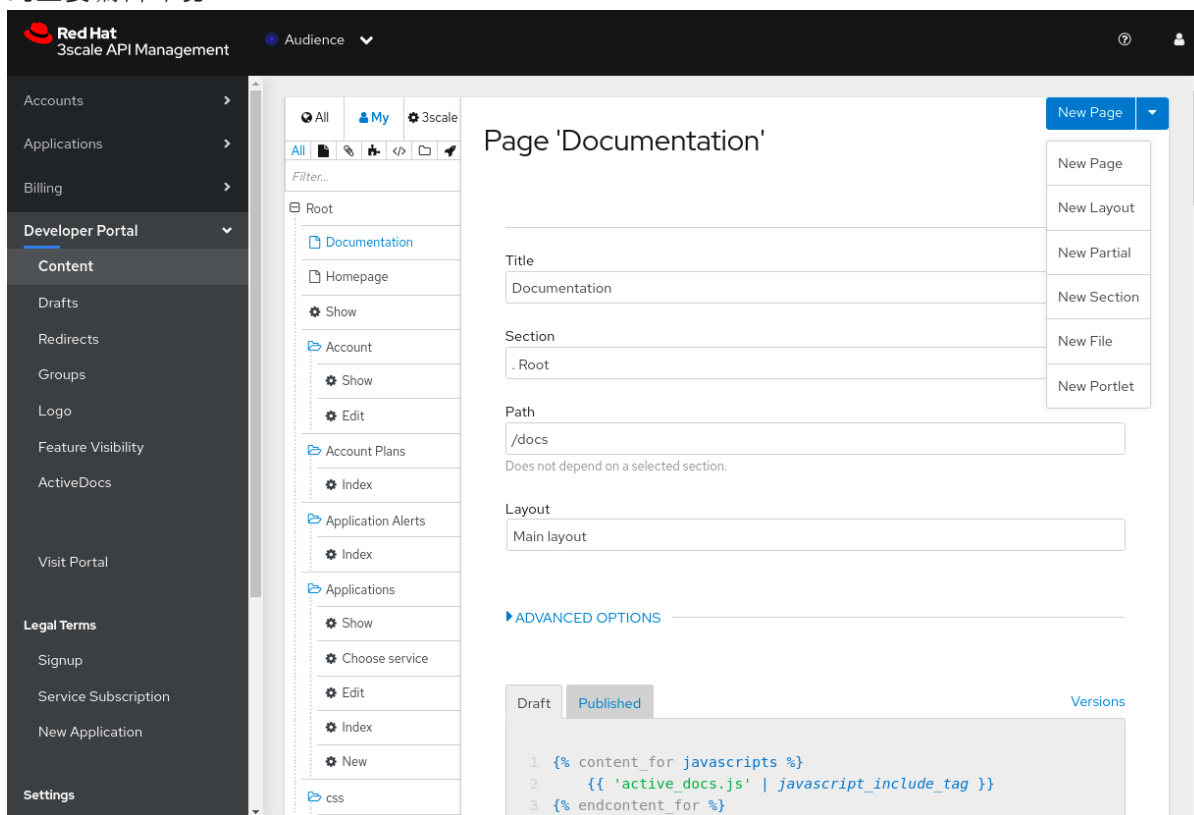
了解和修改原生开发人员门户没有先决条件。但是，在修改原生开发人员门户以创建自己的开发人员门户后，还必须在为 API 用户打开开发人员门户前实施单点登录 workflows 和验证。

1.1. 准备 3SCALE API 管理编辑环境以创建开发人员门户

在开始创建开发人员门户前，请先浏览 3scale 提供的 Echo API 开发人员门户示例。Echo API 开发人员门户是创建自己的开发人员门户的起点。您不会从头开始创建开发者门户。相反，您可以修改原生 Echo API 开发人员门户，以创建具有您想要的外观和感觉的开发者门户。

流程

1. 在 3scale 管理门户中，展开顶部的上下文选择器，再单击 **Audience**。
2. 在左侧的导航树中，展开 **Developer Portal**，再单击 **Content**。这显示了用于创建开发人员门户的主要编辑环境：



在 **Root** 下，3scale 显示 Developer Portal 资源层次结构：

- a. **文档、主页** 和 **Show** 是开发人员门户的基础页面。对于每个页面，向下滚动以查看定义页面内容的 HTML。

- b. 这些页面下的文件夹包含将 3scale 资源拉取到开发者门户的页面。例如，**Account** 文件夹包含用于显示并编辑 3scale 帐户的页面，该帐户时在管理门户中创建的 3scale 管理员。使用这些页面作为起点，并根据需要对其进行修改。
 - c. 在右上角，通过 **New Page** 下拉菜单可以添加一个页面、布局、部分、部分、文件或 portlet。选择每个选项查看您提供的信息来创建。
3. 在选择了 **Developer Portal > Content** 时，将资源层次结构向下滚动到几乎底部，直到您看到 **Layouts** 并点 **Main layout**。
 4. 向上滚动到顶部，可以看到 **Layout 'Main Layout'** 标题。
在内部标题和系统名称后，您可以看到 **Liquid** 已启用。liquid 是 3scale 用于显示和处理 3scale 系统中的大部分数据的框架。定义页面内容的代码包含 Liquid 标记和 HTML。您可以在 **Draft** 选项卡中看到它，其中包含 Developer Portal 页的主布局的代码。
 5. 在左侧的导航树中，在 **Developer Portal > Content** 下的导航树中，点每个子类别来探索它：**Drafts, Redirects, Groups, Logo, Feature Visibility**, 和 **Active Docs**。
 6. 在左侧的导航树中，单击 **Developer Portal** 下的导航树，单击 **Visit Portal**，这是最后一个条目。
新的浏览器选项卡显示 3scale 提供的 Echo API 开发者门户的开发版本的网站。您可以使用此开发版本来查看 3scale 原生开发人员门户。然后，您可以迭代地自定义和查看对原生开发人员门户所做的更改，以创建自己的开发人员门户。

您知道，这是开发人员门户的开发版本，因为它在顶端的 **Draft|Published** 右侧有一个深灰色的面板。Draft 视图支持迭代/递增改进。当 Draft 版本查找并的行为方式时，您可以发布它。

右侧的面板列出了提供当前页面内容的元素：

- a. **页面主页**
 - b. **Layout Main layout**
 - c. **Partial Submenu**
 - d. **Partial analytics**
7. 点 **Page Homepage**。新的浏览器标签页会显示 Developer Portal 编辑环境，其中包含 **Page 'Homepage'** 用于编辑。
 8. 返回到 Echo API 开发人员门户的开发版本，再单击右上角的 **SIGN IN**，它会显示 API 用户用来登录到您的开发人员门户的 **SIGN IN** 页面。
在右边的灰色面板中，模板列表下方，是一个用户名和密码，可用于模拟使用开发人员门户的签名：
 - a. 在 **SIGN IN** 页面中，在 **USERNAME OR EMAIL** 字段中输入 **John**，这是右侧灰色面板中列出的用户名。
 - b. 在 **PASSWORD** 字段中，输入 **123456**，这是灰色面板中列出的密码。
 - c. 点 **Sign In** 显示 Developer Portal，因为您的 API 用户会看到它。

后续步骤

继续探索原生开发人员门户，如您想要的那样。当您熟悉编辑环境和 Echo API 开发人员门户的开发版本时，请遵循 [修改 3scale 原生开发人员门户](#) 的步骤。

1.2. 修改 3SCALE API 管理原生开发人员门户

探索 Echo API 开发人员门户后，在开始创建自己的开发人员门户前对它进行一些更改。这些实践步骤有助于为开发人员门户创建准备工作。

此流程将示例 3scale Echo API 登录页标题替换为通用 Swagger Petstore API 的标题。它还介绍了如何更新开发人员门户来显示 Petstore API 的文档。

流程

1. 在 3scale 管理门户中，展开顶部的上下文选择器，再单击 **Audience**。
2. 在左侧的导航树中，展开 **Developer Portal**，再单击 **Content**。
3. 在 **Root** 下，单击 **Homepage**，它是 Developer Portal 登录页面的内部标题，并更改 Developer Portal 开发版本中显示的登录页面标题：
 - a. 在 **Page 'Homepage'** 中，向下滚动以查看呈现登录页面的代码。
 - b. 在第 **5** 行，更改

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

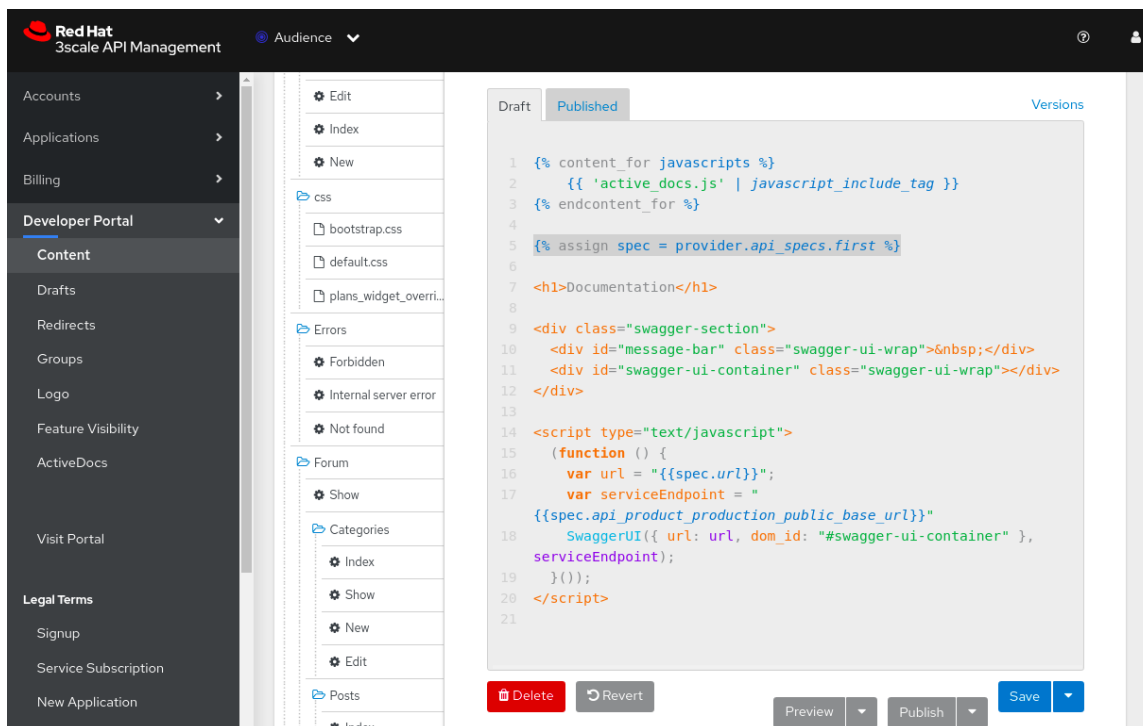
至

```
<h1>Petstore API</h1>
```
 - c. 在页面底部，单击 **Publish**。
 - d. 在左侧的导航树中，单击 **Developer Portal** 下的导航树，单击 **Visit Portal** 以显示开发人员门户的开发版本，并可以看到登录页面标题现在是 **Petstore API**。
4. 保留开发人员门户的开发版本，再单击位于顶部菜单栏中的 **Documentation**。Developer Portal 为 Echo API 显示 ActiveDocs。
5. 返回到 3scale 管理门户并选择 **Developer Portal > ActiveDocs** 查看 Echo API 的条目。3scale 提供定义 Echo API 的 OpenAPI 文档。3scale 使用这个 OpenAPI 文档为 Echo API 显示 ActiveDocs。
6. 导入定义 Swagger Petstore API 的 OpenAPI 文档：
 - a. 进入 <https://petstore.swagger.io/v2/swagger.json>，并将 JSON 内容复制到您的剪贴板中。
 - b. 返回到带有 **Developer Portal > ActiveDocs** 的 3scale 管理门户。
 - c. 在 **ActiveDocs** 页面中，单击 **Create a new spec**。
 - d. 在 **Name** 字段中输入 **Petstore**。
 - e. 选择 **Publish?**。
 - f. 单击 **API JSON Spec** 窗口，粘贴剪贴板中的 Swagger Petstore JSON 内容。
 - g. 点页面底部的 **Create Spec**。3scale 为 Petstore API 显示 ActiveDocs。

- h. 在左侧的导航树中，单击**开发人员门户**下的 **ActiveDocs**。在 **Echo API** 之后，会有一个 **Petstore** 的第二个条目。
7. 在 Developer Portal 中显示 Petstore API 的文档：
 - a. 在左侧的导航树中，单击**开发人员门户**下的**内容**。
 - b. 在 **Root** 下，单击 **Documentation**，这是 Developer Portal 文档页面的内部标题。
 - c. 在 **Page 'Documentation'** 中，向下滚动以查看呈现文档登录页面的代码。第 **5** 行标识开发者门户显示 ActiveDocs 的 OpenAPI 文档。第 **5** 行的默认值为：

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

默认行为是，您的开发人员门户在 **Developer Portal > ActiveDocs** 页面中显示 ActiveDocs 的 ActiveDocs，最初是 Echo API。此图突出显示了第 **5** 行：



- d. 修改第 **5** 行，以将 **provider.api_specs.first** 更改为一个索引，用于识别 ActiveDocs 页面中的第二个条目：


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

默认行为是，您的开发者门户只会显示 ActiveDocs 文档。要显示多个 OpenAPI 文档的 ActiveDocs，您必须在这个简单更改之外修改 **Documentation** 页面。
 - e. 在页面底部，单击 **Publish**。
8. 在左侧的导航树中，单击 **Developer Portal** 下的导航树，单击 **Visit Portal** 以显示开发人员门户的开发版本。
 9. 在顶部菜单栏中，点 **Documentation** 查看 Swagger Petstore 文档。

后续步骤

通过更改原生的 Developer Portal 页面来创建开发人员门户，以便它们显示有关对 3scale 管理的 API 和 API 文档的访问信息。

1.3. 修改原生开发人员门户的其他 3SCALE API 管理功能描述

3scale 提供了许多用于自定义原生开发人员门户的功能，以创建自己的开发人员门户。

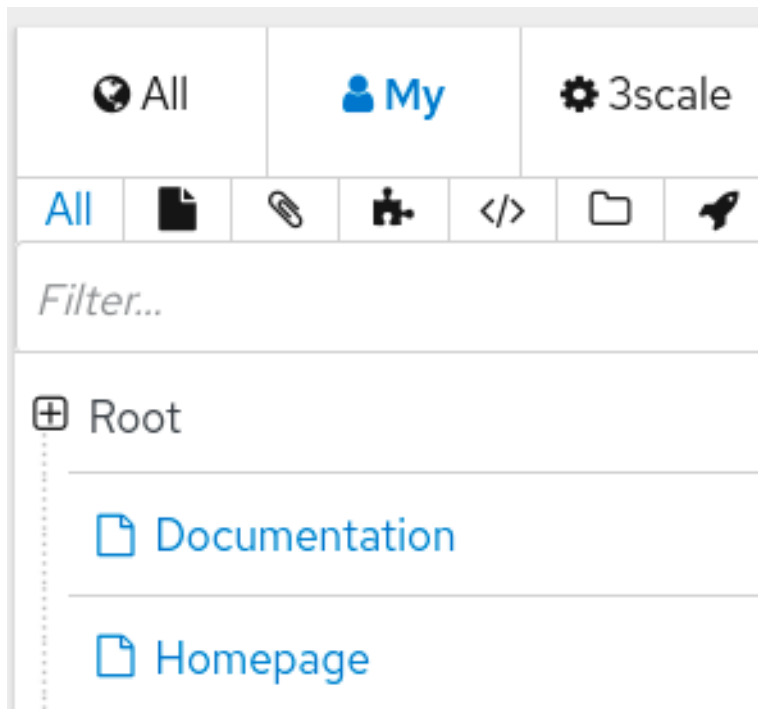
在开发开发人员门户期间，需要查看开发人员门户的任何人都必须指定访问代码。在 API 供应商执行必要的任务来打开您的开发者门户时，请保留此访问代码，但 API 用户访问代码可确保只有具有此代码的任何人都可以查看您的开发人员门户。访问代码位于 **Domains & Access** 页面中，该页面中可通过选择 **Audience > Developer Portal > Domains & Access** 来获得。

以下列表引入了一些修改原生开发人员门户的功能：

- 在 **Developer Portal > Content** 环境中，右上角的 **New Page** 下拉菜单可让您在 Developer Portal 中添加以下元素：
 - **Page** - 开发者门户内容的基本单元。
 - **Layout** - 多个页面可使用的模板。
 - **Partial** - 可在其他部分、布局 and 页面中重复使用的内容。
 - **Section** - 功能相当于一个目录。创建部分以组织开发人员门户内容。
 - **文件** - 您希望开发人员门户使用的资源，如样式表、镜像或脚本。
 - **Portlet** - 外部 RSS 源、内容表或最新论坛文章。
- 您可以在 HTML、Markdown 或 Textile 中输入代码。代码编辑器支持代码突出显示、表格化、行号和其他功能。
- 您可以预览页面的草稿版本或发布版本。在页面的文本编辑器中，单击 **Preview** 会显示开发人员门户的开发版本。
 preview 右侧有一个暗灰色的垂直栏。在黑色星形图的顶部，突出显示 **Draft | Published** 之一来指示您要查看的版本。这个栏包含到编辑环境的链接：
 - 该页面。
 - 页面使用的布局。
 - 页面使用的任何部分。部分代码是在不同页面上的多个位置重复使用的代码。

开发人员门户的开发版本的 Draft 视图支持迭代/递增改进。当 Draft 版本外观和行为时，您可以发布它。当访问代码就位时，发布页面意味着您可以查看 API 用户是否活跃。当访问代码不再就位时，在 live Developer Portal 中发布该页面。

- **filter** 字段允许您在 Developer Portal 环境中搜索资源。3scale 仅显示您要搜索的元素，这样可以更轻松地查找您想要进行的操作。过滤器字段在图标下方：



- **Developer Portal > Redirects** 页面可让您将重定向从一个开发者门户 URL 设置为另一个。例如，如果您弃用了您创建的面，您可以将请求重定向到新页面。
- 占位符是可编辑的字段。例如，**Developer Portal > Content > Login > New** 页面有几个占位符：

```
<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">
```

此代码在 Developer Portal 中生成以下字段：

USERNAME OR EMAIL	<input type="text" value="Authenticate with your email"/>
PASSWORD	<input type="password" value="...and password"/>

您可以更新占位符文本，并发布它来查看开发人员门户中的更新。例如，您可以将 **带有电子邮件的 Authenticate** 更改为 **Authenticate email**。发布页面后，您可以在 Developer Portal 中看到更新的提示，例如：

USERNAME OR EMAIL	<input type="text" value="Authenticate email"/>
PASSWORD	<input type="password" value="...and password"/>

1.4. 3SCALE API 管理布局和部分重复利用开发人员门户内容

在 **Developer Portal > Content** 环境中，当您在资源层次结构中向下滚动时，您可以看到 **Layouts** 和 **Partials** 的标题。布局和部分可让您在开发人员门户中重复使用内容：

- 布局将页面的基本结构定义为页面模板。使用特定布局的每个页面都包含布局定义的内容。正如您在 **布局** 下资源层次结构中所看到的那样，您的原生开发人员门户提供了一个 **错误布局** 和 **主布局**。您可以修改这些布局的代码，您可以创建布局。
- 您在开发人员门户中的多个位置使用的部分代码。例如，部分可以在多个页面上定义所有布局或侧边栏。您可以在页面、布局、其他部分或电子邮件模板中使用部分内容。在资源层次结构中，在 **Partials** 下，您可以看到原生开发人员门户提供的部分。有几部分用于收集分析、订阅应用程序计划、显示子菜单等。同样，您可以修改这些部分，您可以创建部分。在页中，使用部分布局、部分或电子邮件模板来指定 **include** 命令以及部分名称。例如，要使用子菜单部分，您可以指定：

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

布局和部分布局也有草案和发布状态以及版本历史记录。例如，您可以保存草案，发布案例，然后恢复到上次发布的版本。

要了解编辑和编写布局和部分代码的信息，请参阅 [3scale Liquid Reference](#)。

1.5. 在开发者门户中为 3SCALE 管理的 API 添加镜像和其他资产

通常，您需要使用自己的镜像、文件或其他资产来自定义原生开发人员门户。要做到这一点，将文件添加到 Developer Portal 内容库中，记录此环境中的路径，然后在代码中添加内容库中文件位置的链接。

先决条件

- 访问 Developer Portal 中要使用的镜像、文件或其他资产

流程

1. 在 Developer Portal 内容库中添加资产：
 - a. 在 **Developer Portal > Content** 环境中，展开 **New Page** 并选择 **New File**。
 - b. 在 **Upload File** 下，单击 **Section** 字段，以显示您可以进入的目录的列表，然后单击相应的目录来选择它。例如，您可以选择 **|- images**。
 - c. 可选。在 **Path** 字段中，输入对您有用的任何额外路径级别。
 - d. 选择 **Downloadable**。
 - e. 单击 **Choose File**，导航到要添加的文件，选择它，再单击 **Open**。
 - f. 可选。在 **Tag 列表** 字段中，输入您要用来在代码中包含此新内容的 Liquid 标签，例如，输入 **my-logo**。
 - g. 点 **Create File** 将资产添加到 Developer Portal 内容库中。
2. 记录下您所添加资产的路径。例如，如果您将 **MyLogo.png** 镜像添加到 **images** 部分，则路径为 **/images/MyLogo.png**。

3. 要在代码中使用资产，请插入 HTML `<a>` 元素。例如：

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

另外，如果您为资产添加了 Liquid 标签，您可以通过指定标签在代码中使用资产，例如：

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

1.6. 为 3SCALE 管理的 API 自定义原生开发人员门户的注意事项

在开始自定义原生开发人员门户以创建自己的开发人员门户前，您应该为基本元素、可重复使用的内容、页层次结构、页标题和品牌以及品牌制定计划。

- 基本开发人员门户元素 - 开发人员门户的计划应该包括：
 - 站点地图：如何组织门户的框架。
 - 顶部菜单栏中：每个页面上重复的导航。
 - 侧边菜单栏中：访问每个部分中的各个页面。
 - 页面布局准则：了解一致的外观和感觉。
 - 可重复利用的内容 - 布局和部分确保开发人员门户网站的一致性。在开始创建页面前创建它们。
- 页面层次结构 - 从您的站点映射中的根级别开始，为每个顶部菜单项创建一个部分。接下来，创建每个顶部菜单项所需的页面。当您创建页面时，请确保将其与正确的部分相关联。这会在 Developer Portal 中为 URL 路径创建一致的结构。要使用标记语言（如 Textile 或 Markdown）对页面进行编码，请展开页面 **ADVANCED OPTIONS**。Handler 字段标识标记语言。
- 页面标头和页脚 - 通常会在部分定义页面元素。当您只有几个布局时，您可能需要直接在布局中定义标头和页脚。
- 品牌 - 默认原生开发人员门户风格表 **default.css** 非常复杂。使用自定义（覆盖默认值）创建您自己的风格表，而不是扩展此样式表。您可以像创建页面一样创建新的风格表。在高级页面设置中选择 **css** 部分和适当的 MIME 内容类型。然后，在您的布局中，链接到 **default.css** 后，向您的自定义风格表添加一个链接。例如，如果您创建 **my-custom.css**，您可以插入：

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

- 您无法修改的内置功能包括：
 - 分页.
 - 消息菜单.
 - Forums - 您只能更改布局。
 - 付款 - 这些页面中只有几个文本字段可编辑。

1.7. 对于 3SCALE 管理的 API 允许访问开发人员门户的要求

在授予 API 使用者访问开发人员门户之前，API 供应商必须执行以下任务：这些任务可同时执行：

- 将 3scale 原生开发人员门户更改为具有您想要的外观和感觉：
 - [Liquids : 开发人员门户](#) 演示了如何使用 Liquid 标记来显示和处理与 API 相关的 3scale 系统数据。liquid markup 是在 Developer Portal 页面添加逻辑的主要方法。
 - [自定义开发人员门户布局](#) 描述了如何更改原生开发人员门户使其与您自己的品牌匹配。提供标准级联风格表(CSS)，为您的自定义提供简单的起点。
 - [更改内置页面](#) 解释了如何使用 CSS 和 JavaScript 在系统生成的页面中修改或隐藏一个元素。作为原生开发人员门户的一部分提供的所有页面称为系统生成的页面。

熟悉 HTML、CSS、Liquid 和 websites 的开发人员可修改原生 Echo API 开发人员门户以创建您的开发人员门户。此开发人员可在系统生成的页面中创建页面和修改代码，以创建您希望 API 用户在开发人员门户中看到的任何内容。

- 定义 3scale API 产品、后端和应用程序计划，并为产品分配策略。
3scale 产品是 3scale 资源，使您的 API 公开给消费者。在 Developer Portal 中，API 使用者读取产品文档并订阅使用产品提供的 API。3scale 产品有以下特征：
 - 捆绑一个或多个后端，它们是您创建的内部 3scale API。
 - 具有定义关于限制、定价和可用功能的产品使用规则的应用计划。
 - 具有 APIcast 网关配置，用于在向 API 发送 API 使用者调用前如何处理 API 使用者调用。您添加到产品的策略会改变默认的 APIcast 网关行为。

请参阅[管理门户指南](#)。

- 导入定义并记录 3scale 管理的 API 的 OpenAPI 文档。
开发人员门户的基础是一个 OpenAPI 文档，用于定义您的 API。当您把 OpenAPI 文档导入到 3scale 中时，3scale 会创建或更新 ActiveDoc，以便立即为您的 API 功能文档。在 Developer Portal 中，API 使用者使用本文档来探索、测试并与 API 集成。

对于 OpenAPI 文档中定义的每个操作，导入文档会导致 3scale 创建方法和映射规则。方法和映射规则有助于强制强制对 API 使用者访问的限制和规则。

[开发者门户中提供 API](#) 的 3scale 指南包含使用 3scale 和 OpenAPI 文档的信息和流程。特别是，请参阅 [如何编写 OpenAPI 文档以用作 3scale 规格](#)，并将 [ActiveDocs 添加到 3scale](#)。

- 配置 API 使用者如何注册以访问 3scale 管理的 API 的工作流。
使用您的开发人员门户，注册工作流是 API 消费体验的关键方面。工作流可从自助式服务到对谁获得访问权限的全权控制。帐户、服务和应用计划提供多种级别的粒度。在每个级别，您可以控制是否有您操作的批准授权，以及是否需要 API 消费者作出任何选择。

要获得最大自动化和自助服务，您可以删除所有批准步骤，并启用所有可能的默认计划。在注册后，您的开发人员门户可能会发布一个密钥，它提供您的开发人员门户的 API 使用者访问权限。

原生开发人员门户为用户提供、帐户和应用程序注册的常用字段。您可能需要在这些常用的字段中添加自定义字段。详情请参阅[配置注册流](#)和[自定义注册表单字段](#)。在自定义电子邮件模板前，请务必配置工作流。

- 实施 API 使用者身份验证。
对开发者门户的 API 使用者访问权限进行身份验证可保护开发人员门户资源和您的 API。您可以使用以下方法验证对开发人员门户的访问：

- [用户名、电子邮件和密码](#)
- [GitHub](#)
- [Auth0](#)
- [Red Hat Single Sign-On \(RH-SSO\)](#)

请参阅[开发人员门户身份验证](#)。

- 自定义 3scale 原生模板，以获取开发人员门户和 API 用户之间的电子邮件通信。
许多不同的事件需要开发人员门户和 API 用户之间的通信，包括在注册、密码恢复、服务费用、更改通知等时提供帐户激活链接。3scale 提供开发人员门户发送到 API 用户的每个典型电子邮件类型的模板。

在定义了注册工作流后，自定义电子邮件内容。这可让您与为开发者门户设置的工作流匹配。

请参阅 [电子邮件模板](#) 和 [Liquids: 电子邮件模板](#)。

- 指定 API 使用者必须同意访问 3scale 管理的 API 的条款、条件和策略。
当您允许 API 用户注册和调用您的 API 时，您通常希望他们在授予访问权限前就您的条款、条件和策略达成一致。如果开发者门户提供多个服务，您可以有不同的术语和条件版本，例如：注册、使用特定应用程序或使用特定服务。

如果您要使用 API，您可能需要同意您的信用卡策略。

请参阅[设置术语和条件](#)。

- 设置 API 消费账单和信用卡网关。
3scale 计费流程每天运行。它为订阅付费服务的每个 API 使用者帐户创建发票。发票可能处于以下状态之一：open, doned, pending, unpaid, paid, failed, Cancel. 3scale 使用您配置为处理发票的支付网关。

3scale 计费流程可以在预付或后付模式下运行。3scale 中的计费基于日历月份，在每月第一天发生特殊事件。

请参阅 [管理门户指南](#)，[Billing](#)。

在 Developer Portal 变为 live 之前，您必须执行最后的任务是删除访问代码。仅在对开发人员门户进行身份验证进行身份验证后执行此操作，并在您彻底测试了开发人员门户后，以确认其行为已符合预期。

要删除访问代码，显示 **Developer Portal > Content** 环境。在右下角，单击 **Open your Portal to the world** 并确认此操作。

1.8. 为 3SCALE 管理的 API 自定义开发人员门户的可选步骤

除了为 API 用户打开开发人员门户的要求外，您可能需要执行以下操作：

- 在开发者门户中提供多个 API。
您导入到 3scale 的每个 OpenAPI 文档为一个单独的 API 产品提供基础，也称为单独的服务。要将开发人员门户配置为提供多个服务，主要任务是创建一个页面，以便 API 消费者选择要订阅的服务或服务。

详情请查看 [多服务注册](#)。

- 标记 Developer Portal 页面或页面部分，这只对您指定的 API 用户可见。

您可能需要拥有开发人员门户的一部分，只能被特定的 API 用户组访问。您可以限制对页面的访问、页面的一部分或菜单选择（通常对应于某个部分）。

限制访问部分的方便方式是将每个部分映射到 API 用户的逻辑组。例如，假设存在属于合作伙伴的 API 使用者。您可以创建名为合作伙伴的组，并只授予该组对特定部分的访问权限。

您可以根据状态的变化，为 API 用户提供对受限内容的访问。例如，当 API 使用者升级到新的应用程序计划其他页面时，可能会自动对 API 用户可见。

限制访问的另一种方式是需要 API 使用者才能登录来查看某些内容。

详情请查看 [限制内容](#)。

- 实施 webhook。

Webhook 可让您将 3scale 与后台工作流紧密集成。当 3scale 系统中指定事件时，您的后台应用程序可以通过 webhook 消息获得通知。然后，应用程序可以使用这些数据（例如有关新帐户的信息）来填充开发人员门户。

详情请参阅 [Webhooks](#)。

1.9. 更新早于 2.8 的 3SCALE API 管理版本中使用的开发者门户

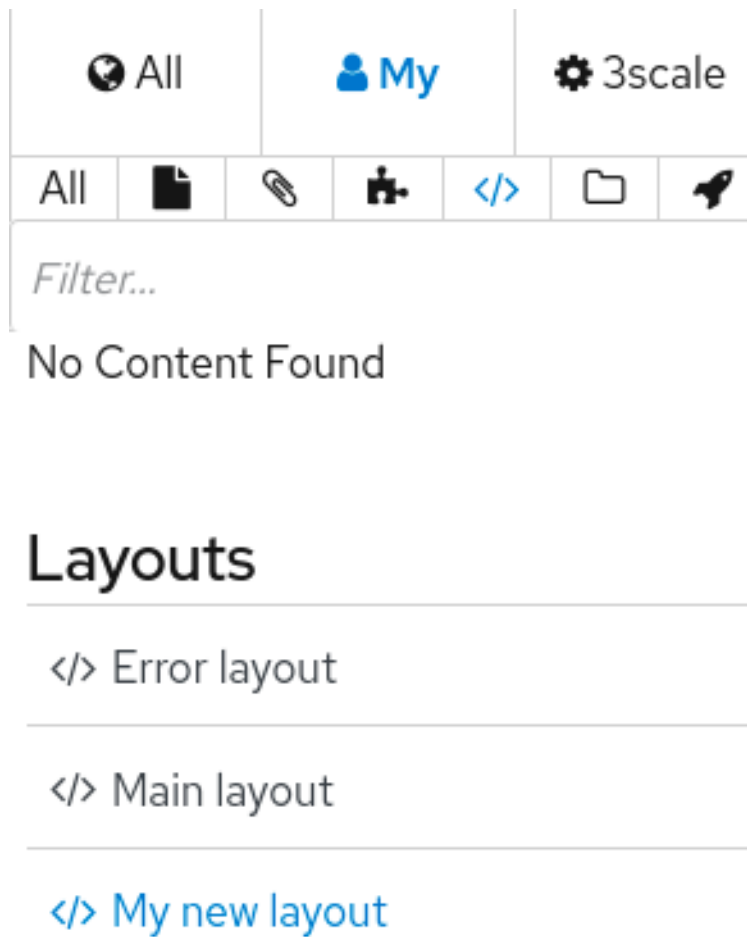
在 3scale 2.8 中，外部资产从 Content Delivery Networks(CDN)移到 3scale 代码库。因此，从 3scale 2.8 开始，原生开发人员门户是使用 `cdn_asset` Liquid 标签创建的。如果要从早于 2.8 的 3scale 版本升级，您必须更新开发人员门户以使用带有 `cdn_asset` 标签的新资产。使用该标签时，从外部网站下载资产不再有依赖项。

先决条件

- 安装 3scale 2.8 或更高版本。
- 使用 3scale 2.7 或更早版本创建的开发者门户。

流程

1. 在 3scale 管理门户中，带有 **Audience > Developer Portal > Content** 在资源层次结构的顶部选择，点 `</>` 仅显示布局：



2. 在 **布局** 下，单击 **主布局**。
3. 在主布局的代码编辑器中，在第 17 行或第 **17** 行替换这一行：

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

使用这个行：

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

4. 在第 **19** 行中，替换此行：

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

使用这个行：

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

5. 向下滚动并单击 **Publish**。
6. 进入资源层次结构的顶部，点  显示部分。
7. 在 **Partials** 下，单击 **stats/chart**。
8. 在 **stats/chart** 的代码编辑器中，在 3 行或第 **3** 行替换这一行：

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

使用这个行：

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

9. 向下滚动并单击 **Publish**。
10. 可选。如果根据主布局或基于 **stats/chart** 部分创建的文件，您必须采用类似更新这些文件。

其他资源

- 3scale Liquid 参考中的 [Liquid drops, tags, and filters](#)

第 2 章 自定义注册表单字段

了解如何添加自定义注册字段以及此功能的不同选项。

默认情况下，3scale 在 `user/account/application` 处提供常用的字段。您可能需要在这些常见默认值中添加自己的自定义字段。

在管理门户中，进入 **Audience > Accounts > Accounts > Field Definitions** 从中可查看默认表单字段并定义新的表单。

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	Edit
----------	---------------------------	----------	------

User

Create


username	"Username"	Required	Edit
email	"Email"	Required	Edit

Application

Create

name	"Name"	Required	Edit
description	"Description"	Required	Edit

新帐户/用户注册页面实际上是前两个部分的修改。帐户字段显示在顶部，后跟用户字段，后跟不需要配置和密码字段。

 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](#)

尝试添加 3 个额外项，2 个添加到注册项，1 个添加到账户项。单击 create，添加以下新字段定义，然后创建它。当然，所需的复选框将使它在报名表上强制使用。另外，还有选项可使内容隐藏且只读。可以添加一个隐藏字段，例如：当您希望新符号设置有字段时，您不希望突出显示它们，如 `access_restricted_are`，它默认为空。作为管理员，您可以逐个用户将其更新为 `true`。您的页面逻辑可将其读取，以确定显示的内容。例如，只读字段可能是浏览器位置，您可以使用 JavaScript on page load 进行设置。

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

现在，尝试在用户注册表单中添加下拉菜单。将它命名为"员工类型"。将这些以逗号分隔的值添加到 choices 字段中：整时、部分时间和合同。下拉菜单将填充这些值。

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.

现在，在帐户中添加预定义的字段。通常，您添加的字段没有系统功能 - 它们只是保存以后可以访问的数据。请参阅 [受限内容](#)。

以正常方式创建字段。然后，在"name"的下拉菜单中，选择 po_number。使用此字段时，发送给此开发人员帐户的 3scale 生成的发票上将显示 PO 编号。系统范围的字段可以随时被您的管理员覆盖。为字段指定名称 - 类似于"PO number" - 并对其进行创建。

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

现在来看一下您的工作。您可以看到免费文本姓氏，并将雇佣类型下拉列表添加到 User 部分。PO 编号系统字段也添加了可用文本。

 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](#))

最后，这些自定义字段可使用 3scale ActiveDocs 进行设置；例如，[应用程序创建](#)。

第 3 章 配置注册流程

在本节中，您将看到将哪些设置配置为调整注册 workflow。

注册 workflow 是您通过开发人员门户提供的开发人员体验的关键方面。此过程可以从完全自动和自助服务到其他极端服务，要求完全控制谁可以访问各种粒度级别。

3scale 平台允许您结合使用帐户（可选）、服务（可选）和应用程序计划对 API 进行建模。对于每个计划，您可以控制是否有您操作的批准。另外，您还要确定是否有默认值，或者开发人员是否需要执行下一步并做出选择。

对于最大自动化和自助服务，请删除所有批准步骤并启用所有可能的默认计划。这样，可以签发密钥以便在注册后立即提供对 API 的访问权限。

3.1. 删除所有批准步骤

要删除批准，请转到 **Audience > Accounts > Usage Rules** 的 **Signup** 部分，确保选中 **Developers** 选项。

The screenshot shows the 'Usage Rules' configuration page in the Red Hat 3scale API Management interface. The left sidebar contains navigation options: Accounts, Listing, Subscriptions, SETTINGS, Usage Rules (selected), and Fields Definitions. The main content area is titled 'Usage Rules' and is divided into sections: 'USER ACCOUNT MANAGEMENT ZONE', 'SIGNUP', 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 the 'Strong passwords' checkbox unchecked.

另外，如果您启用了帐户和服务计划，请向下滚动页面并确保两个情况下都 **直接启用更改计划**：

The screenshot shows the 'Usage Rules' configuration page scrolled down to the 'ADVANCED PLANS' section. The left sidebar is the same as in the previous screenshot. In the 'ADVANCED PLANS' section, the checkbox 'Account Plans' is checked. Below it, the 'ACCOUNT PLANS CHANGING' section has two radio buttons: 'Request a plan change' (unchecked) and 'Change the plan directly' (checked and highlighted with a red box).

3.2. 启用所有可能的默认计划

应用计划

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.

Default Plan: Basic, Unlimited, Plan A, Plan B

Name	Applications	State	Hide	Copy	Delete
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

另外，如果您启用了帐户和服务计划，请选择那些帐户的默认计划。

帐户计划（可选）

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	Publish	Copy	Delete
Default	5	hidden	Publish	Copy	Delete

服务计划（可选）

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

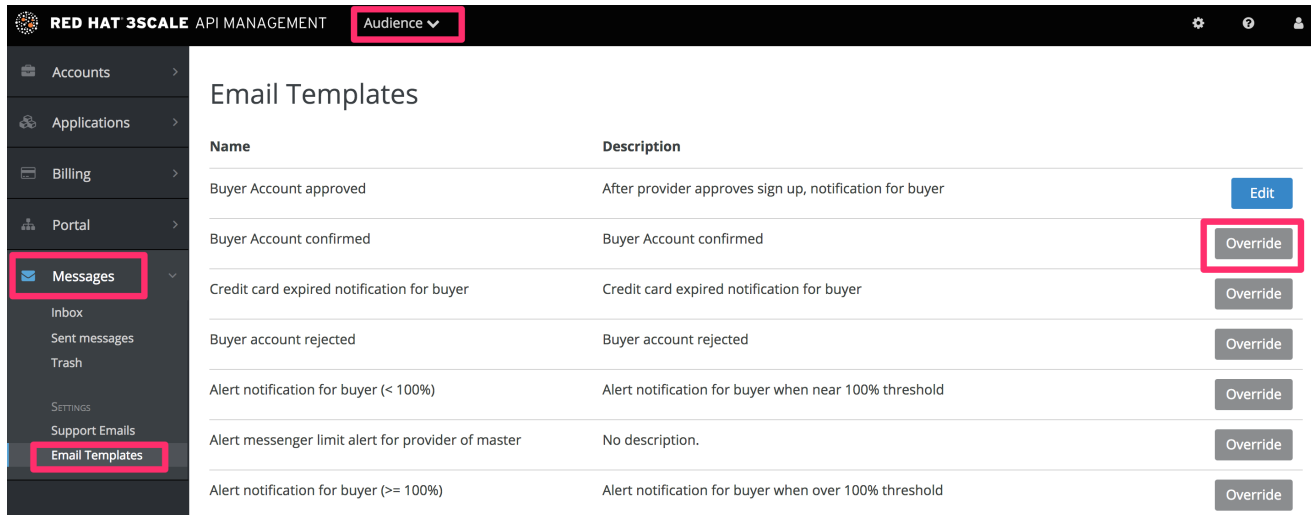
Default service plan (if any) is contracted automatically on sign up.

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

Name	Subscriptions	State	Hide	Copy	Delete
Default	6	published	Hide	Copy	Delete

3.3. 测试 workflow

完成所需的设置更改后，进入开发人员门户并尝试注册新开发人员，以测试结果。试验并做出必要的调整，以准确获取 API 的正确 workflow。当您对工作流满意时，最好检查您的电子邮件通知，以确保他们为开发人员提供正确的信息。



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 多服务注册

在本节结束时，您将熟悉创建和自定义多服务注册页面的步骤。

如果您使用多个服务功能，您可以自定义注册过程来允许客户订阅不同的服务。

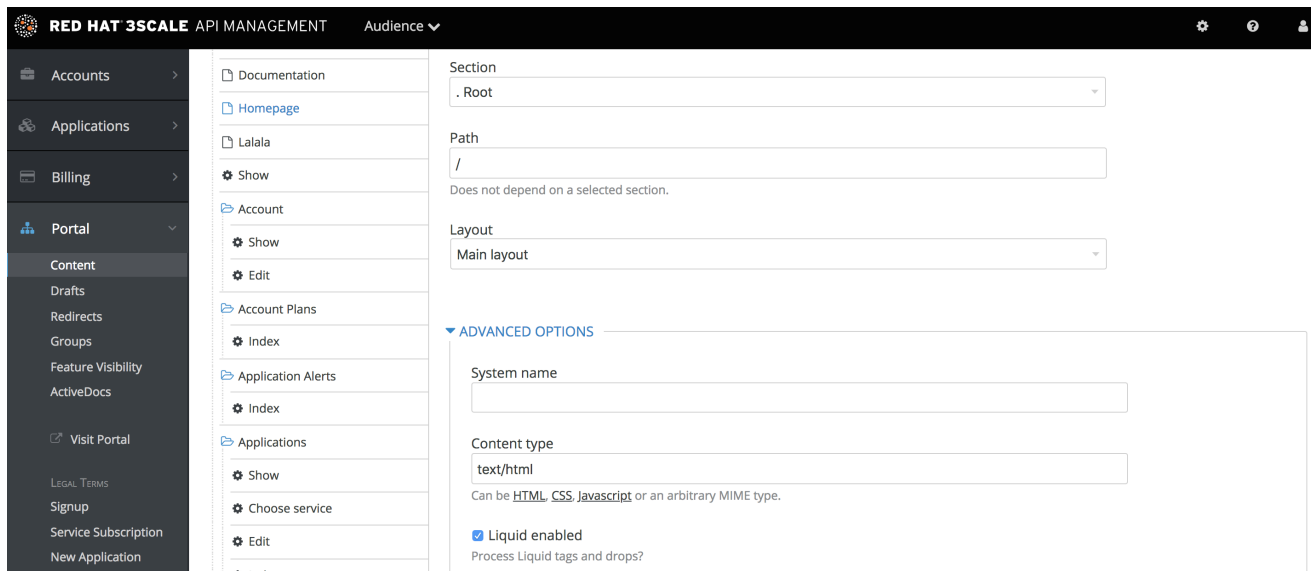
4.1. 先决条件

您应该熟悉布局和页面创建步骤，以及 Liquid 格式标签的基础知识。有关 quid 标签的详情，请参阅 [Liquid 参考](#)。“多个服务”功能必须在您的帐户上启用（可用于 Pro Plan 和 up）。

强烈建议您阅读有关 [注册工作流](#) 的信息，因此您将准备好整个设置，并了解它的工作原理。

4.2. 简介

通过创建新布局来启动该过程，该布局将作为您的多服务注册页面的模板。进入 CMS 系统的布局部分，并创建新的布局。您可以将其称为 *multipleservicesignup*（多服务注册），以方便将其与其他布局区分。在编辑器中，粘贴标准布局（如家或主布局）的一般结构。现在，删除您不需要的一切 - 所有容器、侧边栏、附加框 等。



创建了一个布局的后退性，请继续自定义代码以进行注册。

4.3. 多服务注册

4.3.1. 检索有关服务的信息

要检索有关需要构建正确注册链接的服务的所有信息，您必须通过服务对象进行循环。服务是模型对象的一部分。

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

4.3.2. 配置注册列

您已有布局和循环访问服务对象。现在，决定您要如何显示服务以及注册链接的信息。例如，通过服务描述和底部的注册链接将它们划分为列。每列将作为一门服务列框，其中包含所有必要的信息。

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

内部容器充当自定义 description 字段。service.name 是服务名称，在本例中是容器的名称。

4.3.3. 配置订阅

现在，自定义服务注册的主要部分 - 创建注册链接，提取注册 URL 和服务 ID。从 URL 的对象和服务 ID 中获取您迭代循环的服务对象的注册 URL。最终的链接代码类似如下：

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

您还必须考虑该用户可能已对某些服务进行了签名。创建要检查的条件块。

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

因此，您可以生成最终代码：

```
{% 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. styling

根据您拥有的服务数量，向生成的标记添加一些最终的接触。在本例中，它有两个，因此 service-column div 的 CSS 代码将是：

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



```
.service-column:first-child {  
  margin-left: 0;  
}
```

在示例中，我们使用了基于百分比的布局动态分配包含不v 的尺寸上列的宽度。

现在，您应该有一个正常工作并正确忽略多个服务订阅页面。祝贺您！

如果您要以特定顺序显示列，请尝试使用条件表达式(if/else/case)条件服务名称或其他值。

第 5 章 开发者门户身份验证

按照以下步骤配置开发人员门户的访问权限。

本文介绍了如何在开发人员门户上启用和禁用不同类型的身份验证，以使开发人员能够注册或登录。

目前，3scale 支持多种向开发人员门户进行身份验证的方法，具体将在以下部分介绍：

1. [用户名/电子邮件和密码](#)
2. [通过 GitHub 进行身份验证](#)
3. [通过 Auth0 进行身份验证](#)
4. [通过 Red Hat Single Sign-On 进行身份验证](#)

默认情况下，开发人员门户上只能启用一种类型的验证，如果您已在 3scale.net 中进行了签名：

- 用户名/电子邮件和密码.
- 使用 3scale GitHub 应用程序通过 GitHub 进行身份验证 - 只有在 3scale.net 上签名时才会默认启用

注意

较早的 3scale 帐户（在 2015 年 12 月 14 日之前创建）可能需要遵循额外步骤才能启用 GitHub 和 Auth0 身份验证。

如果这适用于您，则需要将以下代码片段添加到登录并注册模板以便以两种形式启用此功能。

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

5.1. 启用和禁用用户名/电子邮件和密码

默认情况下，开发人员门户上启用了用户名/电子邮件和密码身份验证。通常，此处未进行更改，因为这是开发人员创建帐户和登录的标准方法。

但是，在某些情况下，您可能需要删除这个验证类型。要做到这一点，请按以下屏幕截图编辑 **Login > New** 模板：

```

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

```

如果您需要向开发人员门户添加用户名/电子邮件和密码身份验证，只需删除上一步中添加的 liquid 注释标签即可。

5.2. 通过 GITHUB 启用和禁用身份验证

要启用自己的 GitHub 应用，首先您需要创建一个并检索对应的凭据。

您可以通过 GitHub 配置身份验证的方法有两种：

- 使用 3scale GitHub 应用程序（默认为托管 3scale 帐户）
- 使用您自己的 GitHub 应用程序（用于内部安装）

要更改此默认配置，您可以进入 3scale 管理门户，在 **Audience > Developer Portal > SSO Integrations** 中，您会看到以下屏幕：

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

点击 **GitHub** 访问配置屏幕：



[Edit](#)

Published:

Branding: 3scale branded

Authentication Flow: [Test](#)

在此屏幕中，您可以：

1. 在您的开发人员门户上提供或不可用 GitHub 身份验证，只需选中或取消选中 **发布** 框。
2. 选择 3scale 品牌 GitHub 应用或添加您自己的 GitHub 应用 - 3scale GitHub 应用程序默认为启用（发布）。您可以通过单击 **Edit** 并输入 GitHub 中创建 OAuth 应用程序的详细信息（"Client"和"Client secret"），来配置自己的 GitHub 应用。请注意，为了使集成可以与您自己的 GitHub 应用程序正常工作，您应该使用 "Callback URL" 配置 GitHub 应用程序的授权回调 URL，以便在切换到"自定义品牌"选项后（例如 <https://yourdomain.3scale.net/auth/github/callback>）。
3. 测试配置的身份验证流是否按预期工作。

5.3. 通过 AUTH0 启用和禁用身份验证



注意

此功能仅对企业计划提供。

为了让开发人员使用 Auth0 进行身份验证，您首先需要具有有效的 Auth0 订阅。

默认情况下不启用通过 Auth0 进行身份验证。如果要将在 Auth0 帐户与 3scale 结合使用来管理开发人员门户的访问权限，您可以按照以下步骤进行配置：

进入您的 3scale 管理门户，在 **Audience > Developer Portal > SSO Integrations** 中点击 **Auth0**。

The screenshot shows the 'New Auth0 Authentication Provider' configuration page. 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 the 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'.

在此配置屏幕上，您需要添加 Auth0 帐户的详细信息。输入客户端 ID、客户端 secret 和站点后，选中 **Published** 框，再单击 **Create Auth0** 使其在开发人员门户中可用。

5.4. 通过 RED HAT SINGLE SIGN-ON 启用和禁用身份验证



注意

此功能仅对企业计划提供。

Red Hat 单点登录是一个集成的登录解决方案(SSO)，与 3scale 一起使用时，允许您使用任何可用的红帽单点登录和用户联合选项验证您的开发人员。

有关 Red Hat Single Sign-On 版本与 3scale 兼容的信息，请参阅支持的 [配置](#) 页面。

5.4.1. 开始前

在将 Red Hat Single Sign-On 与 3scale 集成前，您必须有一个正常工作的红帽单点登录实例。有关安装说明，请参阅 Red Hat Single Sign-On 文档：[安装 Red Hat Single sign-on 7.2](#)

5.4.2. 配置 RH SSO 以验证开发人员门户

执行以下步骤来配置 Red Hat Single Sign-On：

1. 按照 [Red Hat Single Sign-On 文档](#) 所述创建域。
2. 将客户端添加到客户端并单击**创建**。
3. 使用以下字段和值填写表单：
 - **客户端 ID**：为您的客户端输入所需的名称。
 - **启用**：切换到 **ON**。
 - **许可需要**：切换到 **OFF**。

- **客户端协议**：选择 `openid-connect`。
- **访问类型**：选择 `机密`。
- **标准化流启用**：切换到 **ON**。
- **根 URL**：输入您的 3scale 管理门户 URL。这应该是您用来登录开发人员门户的 URL 地址，例如：<https://yourdomain.3scale.net> 或您的自定义 URL。
- **有效的重定向 URL**：通过 `/*` 再次键入开发人员门户，如下所示：https://yourdomain.3scale.net/*。
所有其他参数都应留空，或切换到 **OFF**。

4. 通过以下步骤获取客户端 secret：

- 前往您刚才创建的客户端。
- 点 **Credentials** 标签页。
- 在 **Client Authenticator** 字段中选择 **Client Id and Secret Account**

5. 配置 `email_verified` mapper。3scale 要求将用户数据的 `email_verified` claim 设置为 `true`。要将“电子邮件验证”用户属性映射到 `email_verified` 声明：

- 转至 **客户端的映射程序** 选项卡。
- 单击 **Add Builtin**。

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

- 选择 **电子邮件验证** 选项，然后单击 **Add selected** 以保存更改。

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"/>

如果您管理 Red Hat Single Sign-On local 数据库中的用户，请确保将该用户的 *Email Verified* 属性设置为 **ON**。

如果使用 [User Federation](#)，则在为 3scale SSO 集成创建的客户端中，您可以通过将令牌名称设置为 **email_verified**，并将声明值设置为 **true** 来配置硬编码声明。

6. (可选) 配置 **org_name** mapper。

当用户在 3scale 中注册时，会要求用户使用 Organization Name 值填写注册表单。为了通过 Red Hat Single Sign-On 透明登录，不需要在开发者门户上填写注册表格，您需要配置一个额外的 **org_name** 映射器：

- 转至 **客户端的映射程序** 选项卡。
- 点 **Create**。
- 按如下所示填写 mapper 参数：
 - **名称**：键入任何所需名称，例如 **org_name**。
 - **许可需要**：切换到 **OFF**。
 - **映射器类型**：选择 *User Attribute*。
 - **用户属性**：键入 *org_name*。
 - **令牌声明名称**：键入 *org_name*。
 - **claim JSON Type**: 选择 *String*。
 - **添加到 ID 令牌**：切换到 **ON**。
 - **添加到访问令牌**：切换到 **ON**。
 - **添加到 userinfo**：切换到 **ON**。
 - **Multivalued**: 切换到 **OFF**。
- 点 **Save**。

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"/>

如果 Red Hat Single Sign-On 中的用户具有 **org_name** 属性，3scale 将自动创建一个帐户。如果没有，则会要求用户在创建帐户之前指示机构名称。另外，也可以创建类型 *硬编码声明* 的映射程序，将机构名称设置为使用 Red Hat Single Sign-On 帐户登录的所有用户的硬编码

值。


- 要测试集成，您需要添加用户。要实现此目的，请导航到 **Users**，单击 **Add user**，再填写所需字段。请注意，当您在 Red Hat 单点登录中创建用户时，电子邮件验证属性(**email_verified**)应设置为 **ON**，否则不会在 3scale 中激活该用户。





使用 Red Hat Single Sign-On 作为身份代理

您可以使用 Red Hat Single Sign-On 作为身份代理，或者将其配置为联合外部数据库。有关如何配置这些的更多信息，请参阅 Red Hat Single Sign-On 文档来了解 [身份代理](#)和[用户联合](#)。

如果您决定使用红帽单点登录作为身份代理，并且希望开发人员能够跳过 RH-SSO 和 3scale 帐户创建步骤，我们建议以下配置：在提供的示例中，我们使用 GitHub 作为我们的身份提供程序。

- 在 Red Hat Single Sign-On 中，在 **身份提供程序** 中配置 GitHub 后，转至名为 **Mappers** 的选项卡，再单击 **Create**。

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"/>	

- 为它指定一个名称，以便可以识别它。
- 在**映射类型**中，选择 *Attribute Importer*。
- 在 **Social Profile JSON Field Path** 中添加公司，它是 GitHub 上属性的名称。
- 在 **User Attribute Name** 中，添加 org_name，这是我们在 Red Hat Single Sign-On 中调用属性的方式。



注意

Red Hat Single Sign-On 需要第一个和姓氏以及电子邮件以及必填项。3scale 需要电子邮件地址、用户名和机构名称。因此，除了为机构名称配置映射程序外，还能够跳过这两个用户，请确保：

- 在 IdP 帐户中，它们会设置其名字和姓氏。
- 在 IdP 帐户中，可以访问其电子邮件地址。例如，在 GitHub 中，如果您将电子邮件地址设置为私有，则不会共享它。

5.4.3. 配置 3scale API 管理以验证开发人员门户

作为 API 提供程序，配置 3scale 以允许使用红帽单点登录进行身份验证。



注意

默认情况下不启用通过 Red Hat Single sign-on 进行身份验证。Red Hat 单点登录仅适用于企业 3scale 帐户，因此您需要要求您的帐户管理器通过 Red Hat Single sign-on 启用身份验证。

先决条件

- 您的企业 3scale 帐户设置为启用 Red Hat 单点登录。
- [配置 Red Hat Single sign-on 后了解以下详情来验证开发人员门户](#) :
 - **客户端**
Red Hat 单点登录中的客户端名称。
 - **客户端 secret**
Red Hat 单点登录中的客户端 secret。
 - **realm**
您的红帽单点登录帐户的 realm 名称和 URL 地址。

流程

1. 在 3scale 管理门户中，选择 **Audience > Developer Portal > SSO Integrations**
2. 点 **Red Hat Single Sign-On**。
3. 指定您在配置 Red Hat Single Sign-On 中配置的 [Red Hat Single Sign-On 客户端](#) 的详情，以验证 [开发人员门户](#) : client、client secret 和 realm。
4. 要保存您的更改，请点击 **Create Red Hat Single Sign-On**。

第 6 章 RED HAT SINGLE SIGN ON FOR DEVELOPER PORTAL

Red Hat Single Sign On 允许您管理多个独立系统的访问控制。按照本指南，您将能够允许登录系统的用户自动登录到您的 3scale 支持的开发人员门户，而无需再次提示您登录。

本文介绍了如何使用您网站的现有用户凭据来自动登录到您的 3scale 支持的开发人员门户。

此功能适用于已拥有其 API 用户（用户名和密码）身份的 API 供应商 - 例如，API 供应商也是身份提供程序。

6.1. 在 3SCALE API 管理平台中创建用户

首先，API 使用者必须在开发人员门户中有一个帐户。您可以使用帐户管理 API 将用户导入到 3scale，或者手动创建它们。在您的管理门户中找到 3scale ActiveDocs 的帐户管理 API，位于您的管理门户，位于右上角的文档（question mark icon(?)）→ 3scale API Docs 部分。

6.2. 请求登录链接

用户存在后，您可以使用 API 请求调用来生成带有内置 SSO 令牌的 URL：

```
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
```

此调用中有 2 个参数：username，用于指定您要请求令牌的人员和 expires_in，这是令牌的有效秒数（默认为 10 分钟）。

您还可以传递额外的参数 redirect_url，该位置在成功登录后重定向用户。这个参数应该采用百分比编码。XML 响应将包含一个 URL，其中包含了一个 secret 令牌：

```
<?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>
```



注意

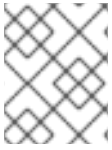
您可以传递 user_id 或 username 来标识 3scale 用户。通常，您的系统和 3scale 门户的用户名相同。在这种情况下，使用用户名应该很容易，因为它不需要将任何附加信息存储在您的一端。但是，如果您需要以任何方式向 URL 进行某种对和机器进程，您可能需要更好地关闭 user_id。

6.3. 使用自动登录重定向用户

响应包含一个带有令牌的 Red Hat Single sign-on 登录 URL：

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

URL 包含 3scale Developer Portal SSO 登录所需的所有信息。您可以直接嵌入 web 中。但是，URL 可以在用户单击之前过期，因此建议您在页面上有一个通用链接，此链接会动态请求新的 SSO URL 并重定向到它。这样，用户就可以无缝登录到开发人员门户。



注意

URL 地址需要取消转义。如果要在浏览器中手动尝试它，请记得在浏览器中将 **&** 替换为 **&**。此外，令牌中的任何 **%** 编码都需要替换为其未转义的字符。

第 7 章 限制内容

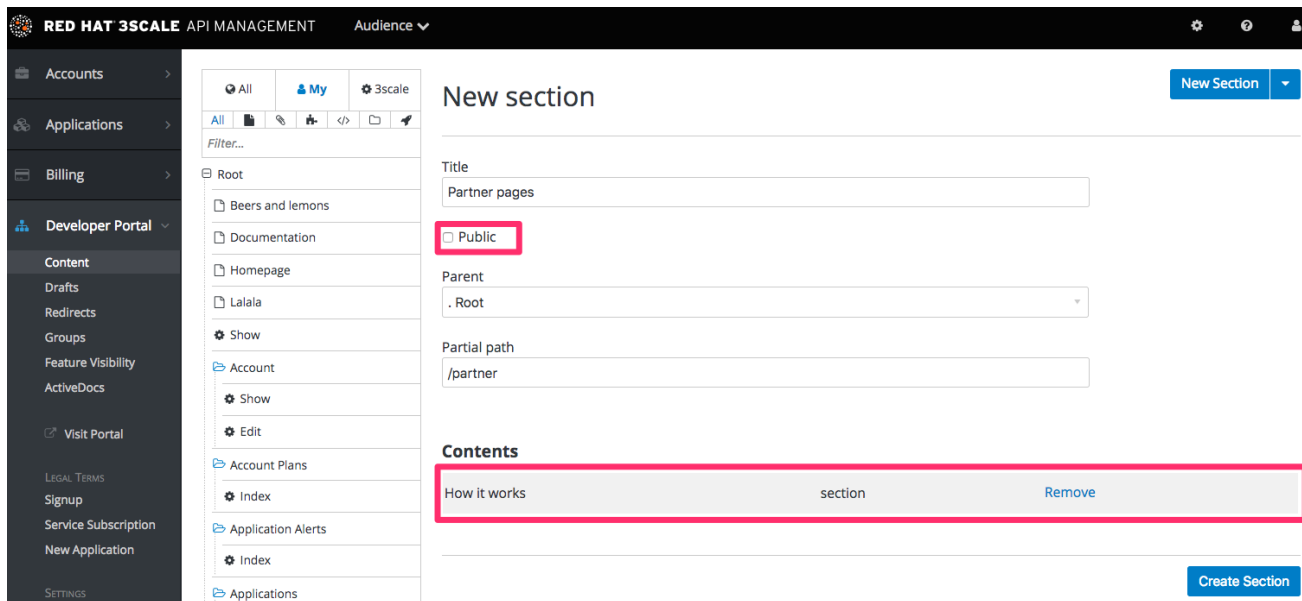
在这里，您可以了解如何在 Developer Portal 中包含仅对某些用户可见的内容。

您可能需要有一些开发者门户的页面，它们只能被一组开发人员（属于某个页的一部分）或某个菜单中的项目访问。这两个目标均可通过以下两个技术来实现。

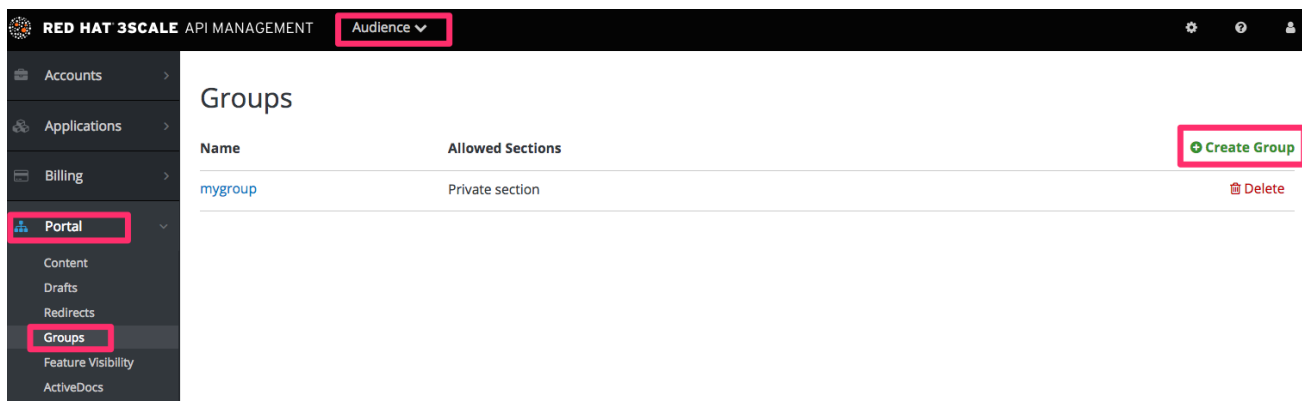
7.1. 受限页面

在创建受限部分时，执行它以便每个部分映射到用户的逻辑组非常有用。在本例中，假设有一组名为“合作伙伴”的开发人员。

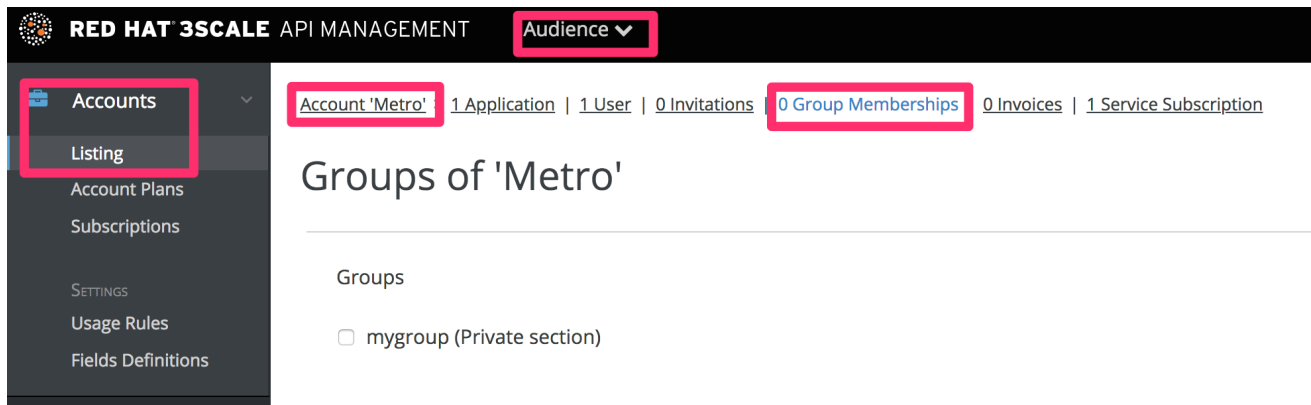
在 Developer Portal 中创建一个新部分，用于您要限制访问的每个页面或一组页面。取消选中“public”状态字段。然后拖放您想要的页面。



创建一个组，并授予它对您创建的部分的访问权限。

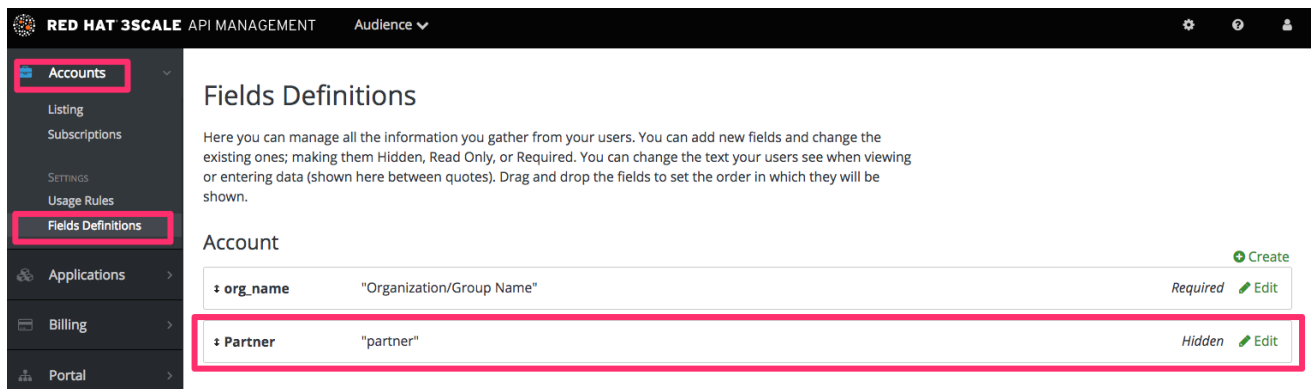


现在，每次您必须授予一个用户对这个部分的访问权限时，您要做的只是将它们分配给这个组。为此，请进入相应的帐户详细信息页面，然后进入“组权限”。一旦进入，请选中您要允许的部分的框。



7.2. 内容的受限块

需要的标记是自定义开发人员门户的一种非常强大方式。请在此处根据条件来隐藏或显示页面的部分。3scale 允许您为帐户、应用程序和用户创建自定义字段。您可以使用它来存储有助于作为 API 提供程序的信息。在这里，您要创建一个附加到所有帐户的自定义字段，并使用它来指明指定帐户是否为合作伙伴。您可以通过转至 **Audience > Account > Field Definitions** 来创建此字段。向 Account 部分添加一个字段，并将它标记为隐藏，这样它就不会显示在注册页面上，或门户上的任何其他位置。



现在您可以使用自定义字段向合作伙伴展示特殊内容，方法是将其嵌套成以下代码片段中的条件：

```

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

```

或者，如果您的情况更好，使用 inverse 逻辑：

```

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

```

在这里，每当您想要向用户显示这些隐藏内容时，您要做的都是在其帐户详细信息页面的"true"字段中键入的。

7.3. 自动配置额外字段

您可以根据状态的变化，为开发人员提供对受限内容的访问。例如，升级应用程序计划的时间。

通过与帐户管理 API 结合使用 [webhook](#) 来简化调配的过程。帐户管理 API 位于 3scale ActiveDocs 中，您的管理门户可提供：

1. 单击 **Documentation**，即窗口右上角的问号图标(?)。
2. 选择 **3scale API 文档**。
3. 通过检查 webhook 请求发送的消息，获取访问受限内容的开发人员的新计划。
4. 根据开发人员的新计划，通过调用 API 更新 `合作伙伴` 字段来授予对私有内容的访问权限。

7.4. 需要用户登录

除了限制对上述内容的访问的两种方法外，还有另一种非常有用的技术：需要已登录的用户。

这非常容易使用 Liquid 标签来实现。您要做的就是包装内容，该内容仅适用于以下条件内的登录用户：

```

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

第 8 章 电子邮件模板

在本节最后，您将编辑并保存自定义电子邮件模板。

您可以完全自定义与开发人员设置的所有标准电子邮件通信的内容，允许您与为开发人员门户设置的工作流匹配。

8.1. 自定义电子邮件模板

8.1.1. 在电子邮件配置前定义您的工作流

有很多电子邮件模板选项，只有一部分与您的工作流相关。在开始编辑电子邮件模板前，请确保您对工作流感到满意，从而节省自己的时间。这样，您将只编辑实际使用的模板。

8.1.2. 测试您的工作流并识别活跃的电子模板

执行完成的工作流的空运行，确保测试所有可能的分支，如批准和拒绝。然后，识别您的测试开发人员帐户接收的每个电子邮件通知，以确定下一步中要编辑的内容。

8.1.3. 编辑并保存您的自定义模板

第一次编辑模板时，您实际上会创建一个自定义模板。然后，在后续的编辑中，您将保存您的更改。



警告

没有版本控制。如果要恢复更改，红帽建议您进行本地副本。

您可以将相关的标签用于电子邮件中的动态内容。特别是在对 quid 标签进行更改时，建议您进行备份。

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. 对工作流中的所有模板重复重复

完成这些步骤，直到您已涵盖工作流的所有可能分支。

8.2. 更多信息

- 在自定义电子邮件模板之前，最好完全完成并测试 [注册流程](#)。
- 如果您要更改电子邮件模板中的任何流动标签，请务必阅读有关 liquid [参考文档](#)。

第 9 章 LIQUIDS : 开发人员门户

本节介绍了 Liquid 格式标签以及如何在 3scale 系统中工作，包括标记的不同元素、它们之间的连接以及如何您的开发者门户中使用它们的简短示例。

要了解有关 Liquids 的基础知识，请参阅 [Liquid 参考](#)。

9.1. 在 DEVELOPER PORTAL 中使用 LIQUIDS

本节介绍如何在布局和页面中启用流动标记处理。

9.1.1. 启用 Liquids

默认情况下，所有部分和电子邮件模板都会启用 quid 标记处理。通过在 system_name 输入字段下的复选框在布局上启用它们。但是，要在页中启用它们，您必须前往该页面的高级选项部分。

▼ 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

只需展开高级选项部分，并标记“启用”复选框。现在，所有 liquid markup 将由内部引擎处理，开发人员门户内置编辑器也会添加代码高亮显示。

9.1.2. 在页面、部分和布局上使用不同的用法

liquids 的使用通常略有不同，页面、部分和布局之间稍有不同。在页面内，liquids 是单使用元素；而与部分和布局相对应的是开发人员门户的可重用元素。这意味着，除了对不同页面进行小更改时不应用多个布局或部分，您也可以添加一些逻辑流动标签，并根据用户所在的页面更改布局。

```
<!-- 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. 与 CSS/JS 一起使用

liquid markup 不仅可与 HTML 配合使用，您可以轻松地将其与 CSS 和/或 JavaScript 代码组合，以便更多控制。若要启用 liquid in a JS，将其创建为页面页面，并遵循与您为普通页面启用相同的步骤。完成后，您将能够在 CSS 中添加一些条件标记，或者在 JavaScript 中使用服务器端数据。只需记得将页面的内容类型设置为 CSS 或 JS。

9.2. 在电子邮件模板中使用 LIQUIDS

本节介绍如何使用相关的标签来自定义电子邮件模板。

9.2.1. Developer Portal 的不同

如前文所述，liquid 标签还可用于自定义发送给用户的电子邮件模板。所有编写时提到的常规规则同样适用于电子邮件模板，但有一些例外：

- 每个模板都没有常用的变量列表。反之，您必须使用前面提到的 `{% debug:help %}` 标签进行一些测试。
- 由于电子邮件本质上与网页不同，因此您将有限或无权访问某些标签。例如，`{{ request.request_uri }}` 不会有意义，因为电子邮件没有 URL。

9.3. 故障排除

此故障排除部分将帮助您调试和修复可能出现的典型错误。

9.3.1. 调试

如果某个操作没有按预期工作，但会正确保存，请检查以下内容：

- 所有标签都被正确关闭。
- 您指的是当前页面中可用的变量。
- 您不尝试访问数组 - 例如 `current_account.applications` 是一系列应用程序。
- 逻辑正确。

9.3.2. 用于解决这些错误的典型错误和方法

- 如果因为 liquid 错误而无法保存文档，则通常是因为一些标签或丢弃没有被正确关闭。检查所有 `{% %}` 和 `{{ }}` 标签是否都正确关闭，并且逻辑表达式，如 `if`, `for` 等，都有相应的 `endif`, `enfor` 通常，如果有问题，编辑器上方页面顶部将显示错误，其中包含一个描述性错误消息。
- 如果一切都正确保存，且您没有看到任何效果，请检查您没有引用空元素，并且您没有使用 `logic` 标签来显示内容。`{% %}` 不会呈现任何内容，除了在标签中使用已经是比较复杂标签和丢弃的别名的标记。
- 如果只会显示 `#` 符号，这表示您已尝试显示是一个数组的元素。检查有关 [liquid 层次结构](#) 的部分。

9.3.3. 联系支持

如果您仍有问题，您可以通过 [红帽客户门户网站](#) 创建一个新问题单。

第 10 章 LIQUIDS : 电子邮件模板

3scale 提供使用您组织自己的消息传递和术语来自定义电子邮件模板的功能。您还可以利用流动的丢弃来显示每个客户的个性化信息。

与 Developer Portal 中的 quid drops 类似，每个电子邮件模板都有自己的上下文。这意味着，一个电子邮件模板中提供的丢弃可能不一定可用于其他电子邮件模板。

本参考概述了在什么地方提供相关的电子邮件模板，电子邮件模板按主题问题分组，以及一组获得它的丢弃丢弃了它们的支持。

10.1. 帐户管理

这些电子邮件模板属于帐户管理类别：

- 已确认购买帐户。
- 购买者帐户已批准。
- 买方帐户被拒绝。

对于这些模板，您可以使用以下 liquid drops：

- **user** ⇒ **User**
- **domain** ⇒ **String**
- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **support_email** ⇒ **String**

另外，以下模板：

- 买方的密码恢复可以访问这些 liquid drops：
- **user** ⇒ **User**
- **provider** ⇒ **Provider**
- **url** ⇒ **url**

向帐户邀请其他用户的电子邮件：

- 邀请可以访问：
- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **url** ⇒ **url**

10.2. 信用卡通知

- 供应商的信用卡过期通知。

- 用于买方的信用卡过期通知。

您可以使用以下 liquid drops :

- **user_account** ⇒ **Account**
- **account** ⇒ **Account**
- **provider_account** ⇒ **Provider**
- **provider** ⇒ **Provider**

10.3. 限制警报

- 提供者的警报通知($\geq 100\%$)
- 为购买者的警报通知($\geq 100\%$)
- 提供者的警报通知($< 100\%$)
- 通知购买者($< 100\%$)

可以访问 :

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

10.4. 应用程序

以下电子邮件模板均处理应用程序和应用程序计划通知。

- 为供应商创建的应用程序可以访问 :
- **url** ⇒ **url**

应用程序计划更改请求通知电子邮件模板 :

- 计划购买者的更改请求。
- 计划更改提供程序请求。

他们可以访问 :

- **application** ⇒ **Application**
- **provider** ⇒ **Provider**
- **account** ⇒ **Account**
- **user** ⇒ **User**

- **plan** ⇒ **Plan**
- **credit_card_url** ⇒ **credit_card_url**

以下电子邮件模板包含多个可用 drop，例如：

- 为购买者更改了应用程序计划。
- 为供应商更改了应用程序计划。
- 购买者已过期应用程序试用期。

他们可以访问：

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

以及上述所有 liquid drops，以下应用程序计划信息：

- 暂停应用程序以购买者。
- 已接受购买者的应用程序。
- 拒绝购买者的应用程序。
- 已取消供应商应用程序合同。

另外列出了 liquid drops：

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

针对应用程序键的以下电子邮件模板会更提示：

- 为购买者创建的应用程序密钥。
- 为购买者删除应用程序密钥。
- **key** ⇒ **key**

10.5. INVOICING

以下电子邮件模板：

- 在为供应商支付前查看发票：
- **provider** ⇒ **Provider**
- **URL && String**

另外，以下模板：

- 没有重试的供应商的发票费用失败。
- 购买者的发票。
- 带有重试的供应商的发票费用失败。
- 对没有重试的买家的发票失败。
- 成功为购买者收取发票。
- 对带有重试的买方的发票失败。

分享以下 liquids :

- **account** ⇒ **Account**
- **provider** ⇒ **Provider**
- **cost** ⇒ **cost**
- **invoice_url** ⇒ **invoice_url**
- **payment_url** ⇒ **payment_url**

10.6. 服务

以下电子邮件模板 :

- 已取消供应商服务合同。
- 购买者已过期服务试用期。
- 为供应商更改了服务计划。
- 为购买者暂停服务合同。

可以访问 :

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

以及上述 liquid drops, 以下服务模板 :

- 为供应商创建的服务。
- 为购买者接受的服务。
- 拒绝购买者的服务。

另外列出了 liquid drops :

- **service** ⇒ **Service**

- **service_contract** ⇒ **Contract**
- **subscription** ⇒ **Contract**

10.7. 注册

以下电子邮件模板：

- 供应商的注册通知。
- 购买者的注册通知。

可以访问：

- **user** ⇒ **User**
- **provider** ⇒ **Provider**
- **url** ⇒ **activate_url**

第 11 章 自定义开发人员门户布局

您可以自定义整个开发人员门户的外观和感觉，使其符合您自己的品牌。提供了标准 CSS 风格表，供您自定义提供简单的起点。要创建布局模板，请将 **主布局** 的代码用作您的起点。

在本教程中，您将把自己的 CSS 自定义添加到开发者门户，并重新载入它以推出新的实时更改。

11.1. 创建新的 CSS 文件

有默认的风格表 **default.css**。这非常大且复杂，而不是扩展它，最好为您自己的自定义创建自己的风格表来覆盖默认值。您可以像创建页面一样创建新风格表。记得在高级页面设置中选择适当的 MIME 内容类型。



重要

确保所选的布局为空。否则，页面布局 HTML 将模糊处理 CSS 规则。

11.2. 将风格表链接到您的页面布局

在链接到 **bootstrap.css** 的链接后，将每个布局模板（或部分有共同的 HEAD 部分）链接到您的自定义 CSS 的链接。例如：

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

现在，享受您自己的独一无二的品牌！

11.3. 定义页面布局模板

常规的概念是为门户中的每个不同页面风格定义单独的布局。在启动时有一个名为 **Main layout** 的标准布局。在使用开发人员门户时，不要对此布局进行任何更改，因为所有系统生成的页面使用此布局。

通常，您希望门户主页的唯一风格。**Main layout** 模板是您自定义的起点。创建页面布局模板：

1. 打开 **主布局** 并将其代码复制到剪贴板。
2. 创建新的布局，为其提供标题、系统名称，然后选择 **Liquid enabled**。
3. 将 **Main layout** 代码粘贴到新的布局中。
4. 通过从您的新布局中删除这一行来删除侧边栏菜单：

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

5. 自定义代码以创建布局模板。

第 12 章 更改内置页面

在本节结束时，您将能够修改并配置系统生成的页面中任何元素的可见性。

系统生成的一些元素无法从 Developer Portal 修改，如 Signup、Dashboard 和 Account 页面。本指南介绍了如何使用 CSS 和 JavaScript 在这些页面上的内容进行自定义。

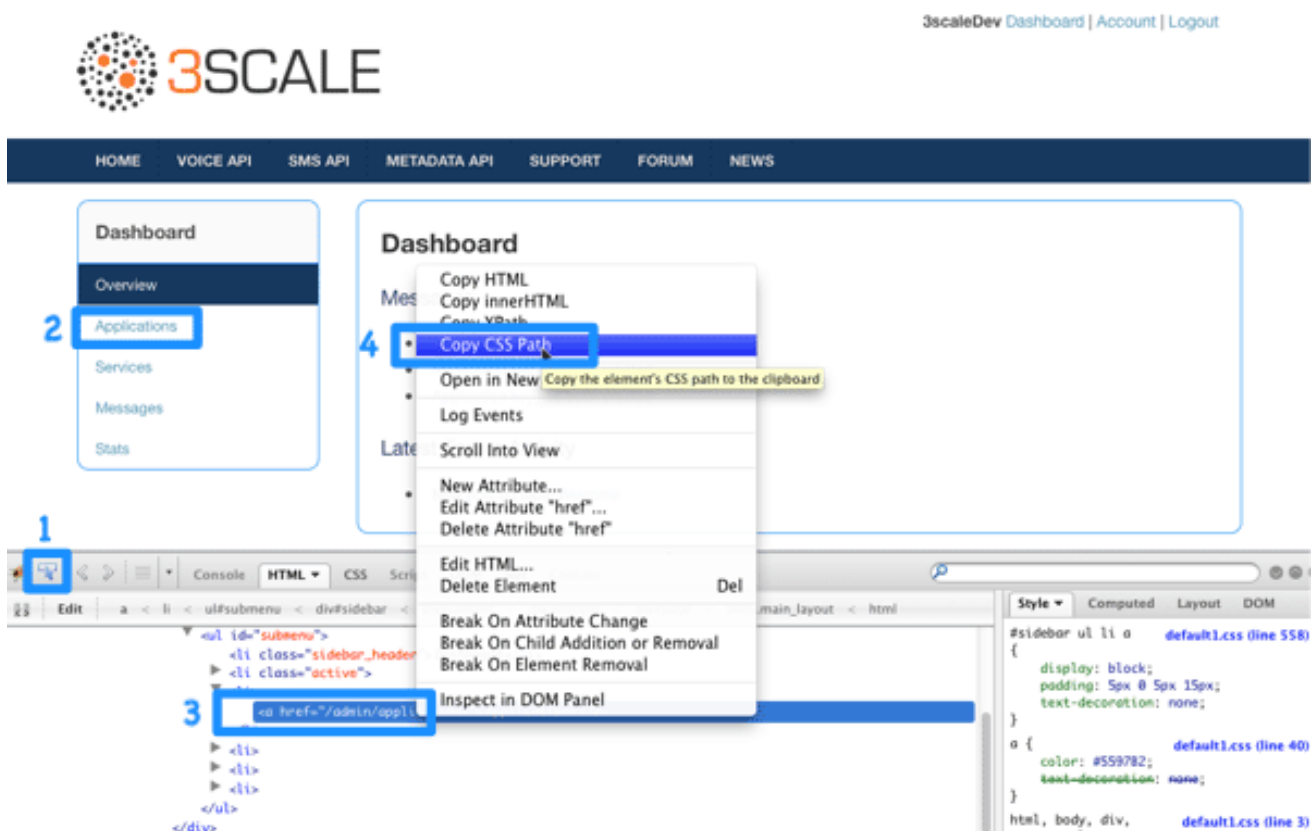
系统范围的页面遵循后端规则访问和可见性，因此其 URL 必须是特定的和硬编码的值。客户门户网站中的知识库文章提供了 [系统生成的页面及其 URL 列表](#)。

小心

3scale 系统生成的页面可能会改变，尽管不常发生。这些更改可能会破坏您在本指南下实施的任何自定义。如果避免使用这些黑色，请这样做。在继续之前，请确保您可以监控任何破坏性更改，并执行必要的维护工作，以保持您的门户正常。

12.1. 确定元素

最重要的一点是确定要隐藏的内容。为此，请使用 Firebug（或其他开发人员工具，如 Chrome 开发人员工具或 Opera Dragonfly）。选择所需元素，并在控制台中点击它并选择 Copy CSS 路径。这样，您可以保存准确的 CSS 路径，以简化操作。请记住，如果元素是侧栏导航小部件的一部分，您还必须指定列表中的哪个位置。对于这一点，您可以使用 "+" 选择器（例如，选择第 3 个部分），例如：`ul + li + li + li + li + li` 或 `:nthchild(n)` CSS3 pseudoclass。



12.2. 修改或隐藏元素

现在，已确定了元素，您可以更改其显示设置。根据元素类型，您可以从以下两种可能的方法中选择：CSS 操作或 jQuery 脚本。CSS 操作更加轻便且更可靠，但对于多个页面上存在的某些种类的元素（例如，管理门户控制面板侧栏的第三句元素也存在不同的值）。某些棘手的实施需要使用 CSS3，而这些方案不受旧浏览器的支持。在接下来的两个步骤中，您将看到这两种方法。

12.3. 选项 A : CSS

例如，尝试在 Dashboard 页面中隐藏最新的元素。第一步，您已将其 CSS 路径识别为：

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

请记住，这是第二个具有相同路径的方框，因此您将使用 "+" 选择器。您的路径现在类似如下：

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

将 display 属性改为 none 会导致这个方框不可可见：

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

12.4. 选项 B: JQUERY

如果您有复杂的元素来隐藏一个侧边栏菜单元素，最好使用一些 jQuery。这些元素的 CSS 路径在 Dashboard 和 Account 部分中是一致的，您不想在这两个部分中隐藏元素。因此，请根据 CSS 路径和内容选择元素。在本例中，假设您想从 Dashboard 的边栏中隐藏 messages 部分。您的 CSS 路径为：

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

要匹配内容，请使用 `.text ()` 函数。另外，在文档头和就绪函数内包括代码，以便在生成所有内容后运行。

The screenshot shows the 3scale Developer Portal interface. The top navigation bar includes 'Dashboard', 'Developers', 'Applications', 'Billing', 'Analytics', 'API', 'Developer Portal', and 'Settings'. The 'Developer Portal' tab is active. The left sidebar has 'Content', 'Redirects', and 'Changes' tabs. Under 'Content', 'main_layout' is selected. The main content area is titled 'Layout 'Main layout'' and contains the following configuration:

- Title: Main layout
- System name: main_layout
- Liquid enabled: Process Liquid tags and drops?

At the bottom, there is a code editor with the following 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 {% scale_essentials %}
18
19 </head>
```

A blue box highlights the JavaScript code in the code editor.

生成的代码片段类似如下：

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

这不是唯一的解决方案。它只是显示这样一种可能的实现方法。同样的示例可以使用纯的 CSS 带有属性值的 CSS3 选择器。有关完整的 CSS3 选择器规范，请 [在此处](#) 查看。

第 13 章 设置条款和条件

当您允许开发人员注册您的 API 时，您可能需要在授予其访问权限前，同意您关于您的条款和条件，以便了解其中的一些策略。

您的条件和条款可能还会有不同版本的开发人员。在注册过程的不同点上很容易设置。例如：

1. 注册条款和条件.
2. 应用程序条款和条件.
3. 服务/订阅条款和条件.
 - 仅在有多个服务时可用。

另外，如果您正在使用 API，您可能要明确将您的信用卡策略设置为明确。3scale 提供了设置以下信用卡策略 URL 类型的简便方法：

1. 法律条款
2. 隐私性
3. 退款

13.1. 条款和条件

该 workflow 部分通过下列步骤在管理门户中轻松设置。

进入 **Audience > Developer Portal > Signup** 在其中会显示一个空白页面来填充您的签字法律条款。您可以使用 HTML、JavaScript 和 CSS 的任意组合。还可以通过单击 Insert toggling 代码来提供一些切换的代码。您在此框中编写的内容只需在开发人员门户的 **Sign Up** 页面上的 Sign Up 按钮上方出现。

The screenshot shows the Red Hat 3scale API Management interface. At the top, the header reads 'RED HAT 3SCALE API MANAGEMENT' with a dropdown menu currently set to 'Audience'. On the left sidebar, the 'Portal' menu item is expanded, and 'LEGAL TERMS' and 'Signup' are highlighted. 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 text area for editing, with the number '1' at the top left.

填写了条款和条件后，点 Update 保存它们。

如果您使用切换代码，它将显示"通过签名，您同意以下法定条款和条件"，后跟一个链接，在显示和隐藏您指定的条件和条款之间切换的链接。

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, there are five input fields: "ORGANIZATION/GROUP NAME", "USERNAME", "EMAIL", "PASSWORD", and "PASSWORD CONFIRMATION". A blue-bordered 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.

默认情况下，这放置在 Signup 页面上，但它是一个部分(signup_licence)，可在开发人员门户的任何位置包含。要从 Signup 页面删除它，只需在页面中删除 `{% include 'signup_licence' %}` 行。同样，如果在其它位置包含它，您可以通过代码片段的其它部分使用相同部分，这些部分可以放置在开发人员门户网站中的任何位置。

您可能还希望用户在创建新应用程序时(`new_application_licence` partial)和/或订阅新服务(`service_subscription_licence` partial)时，用户接受另一组条款和条件。要设置这些设置，您可以按照上述步骤相同的步骤。

13.2. 信用卡策略

您还可以定义不同策略所在的其他 URL。要设置它们，请转至 **Audience > Billing > credit Card Policies** 并设置您的策略页面所在的路径。

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

要使这些链接正常工作，您将需要在 Developer Portal 中创建新页面。

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

完成后，您可以使用 URL 的 liquid drop 引用它们。例如：

```
<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>
```

第 14 章 导出和导入开发人员门户

作为 3scale API 供应商，您可以为以下目的导出和导入开发人员门户：

- 创建备份。
- 将开发人员门户保留在外部存储库中，如 GitHub。
- 将开发人员门户与其他应用程序集成。

使用 Developer Portal API 作为内容管理系统(CMS)导入和导出开发门户内容。为此，请执行以下步骤生成具有足够权限的密钥以使用 Developer Portal API。

流程

1. 导航到 **Account settings > Personal > Tokens**，再点 **Add Access Token**。
2. 将访问令牌命名为，并检查 **Developer Portal API**。
3. 选择权限：
 - a. **只读** 允许检索开发人员内容。
 - b. **读和写** 允许检索和恢复开发人员门户内容。
4. 单击 **Create Access token**。
5. 复制和存储显示的令牌信息。

检查从左侧面板浏览的端点列表，以 **integrate & gt; 3scale API Docs**。然后，向下滚动到 **Developer Portal API**。使用生成的令牌来调用每个端点，并根据您的需要填写字段。

开发者门户 API 端点注意事项

3scale 2.14 提供的开发者门户 API 与之前的版本不兼容。另外，在 3scale 2.14 中，JSON 是唯一兼容所有请求和响应的数据格式。

对于每个端点，您可以执行以下操作：

- **GET** 来读取和列出资源。
- **POST** 用于创建和添加资源。
- **PUT** 来修改资源。
- **DELETE** 删除资源。



注意

- 无法删除内置对象。使用 **type=builtin_page** 参数调用 **GET /admin/api/cms/templates** 端点，以获取内置页面列表和 **type=builtin_partial** 参数，以获取内置部分列表。
- 要进行完整的备份，您必须调用每个内容。没有 API 端点可下载包含所有文件的完整存档。
- 如果没有发送 **内容**，它不会返回发布或草案内容。相反，它会返回一个包含模板名称和部分等信息的摘要，因为内容太长。

在执行后，使用每个端点下列出的详细信息来优化其输出。对于列出的参数，请考虑以下几点：

- 所有端点都拒绝不支持的参数；如果发送不支持的参数，则请求将取消。
- **GET /admin/api/cms/templates** 端点接受 **内容** 参数。默认情况下，它会返回一个 Developer Portal 模板列表。要同时发布和草案内容，请使用 **content=true** 参数。
- **GET /admin/api/cms/templates** 端点接受 **type** 和 **section_id** 参数来过滤结果。
- **GET /admin/api/cms/sections** 端点接受 **parent_id** 参数来过滤结果。
- **GET /admin/api/cms/files** 端点接受 **section_id** 参数来过滤结果。