



Red Hat 3scale API Management 2.14

Liquid 参考

查找与 3scale API 管理安装相关的其他信息。

查找与 3scale API 管理安装相关的其他信息。

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

本指南提供有关与 3scale API 管理安装交互的组件的信息。

目录

对红帽文档提供反馈	3
第 1 章 LIQUID 参考	4
1.1. 什么是 LIQUIDS?	4
1.2. 如何使用 LIQUID	4
1.3. LIQUID 模板	5
1.4. DROPS	6
1.5. TAGS	8
1.6. 过滤器	8
1.7. DROPS (UP)	8
1.8. TAGS (UP)	61
1.9. FILTERS (UP)	66

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 LIQUID 参考

1.1. 什么是 LIQUIDS?

Liquid 是一种简单的编程语言，用于显示和处理 3scale 系统中的大部分数据，供 API 提供商使用。Liquid 最初由 [Shopify](#) 开发，并在整个网络的许多其他内容管理系统(CMS)引擎中使用。在 3scale 平台中，它用于向您的 API 开发人员公开服务器端数据，在维护高级别安全性的同时大幅扩展开发人员门户的实用性。

1.1.1. 优点和缺点

人员主要用于在开发人员门户页面上获取和显示服务器端数据。但是，比这么多。这些服务非常适合：

- 根据服务器端数据更改页面的 DOM 和内容
- 在页面、布局和部分中添加逻辑
- 操作发送给开发人员的电子邮件模板

有些用例中，攻击并不提供问题的最佳解决方案，大多数情况是您需要使用动态数据，如用户输入或页面 URL。

一些一般建议是将它们用作向页面添加逻辑的主要方法，但是当您发现不可能或过于复杂时，切换到 JavaScript（或将其添加到其中，因为与 JS 非常适合）。

1.2. 如何使用 LIQUID

Liquid 标记分为两种类型：逻辑标签和输出标签。标记为 `{% %}` 的逻辑标记是条件语句，包含标准编程语言元素，如 `if` 子句和循环等。

```
{% if current_user %}      <!-- if the user is logged in -->
  <a href="/logout">Logout</a> <!-- show the logout link -->
{% else %}                 <!-- if the user is not logged in -->
  <a href="/login">Login</a> <!-- display the login link -->
{% endif %}
```

输出标签标记为 `{{ }}`，用于显示大括号之间的标签值。

```
{{ current_user.username }} <!-- display the logged-in user's username value -->
```

有关逻辑标签的文档，请参阅 [Shopify 教程](#)。3scale liquid 输出标签的完整参考可在 [tags](#) 部分找到，并在您的管理门户的 **Help > Liquid Reference** 下找到。

1.2.1. Liquid 丢弃、标签和过滤器

在 3scale 开发人员门户中，您可以访问三种类型的 Liquid 标记：

- [liquid drops](#) - `{{ current_account.name }}`
- [liquid tags](#) - `{% content %}`
- [liquid filters](#) - `{{ form.errors.description | error_class }}`

Liquid drops、标签和过滤器与 [How to use liquids](#) 中提到的逻辑和输出标签类似。

Liquid drops 是最基本的结构，您经常会使用它们。它们提供对系统中存储的某些值的访问权限，如用户名和应用的标识符。解释器处理它们的方式与任何其他输出标签相同。

另一方面，Liquid 标签是一种逻辑标签，可以呈现或访问系统的某些部分进行进一步自定义，例如，呈现布局中的内容或自定义电子邮件模板。

通过过滤器，可以选择从丢弃、转换值和按某些键分组的结果。有一组标准的 liquid 过滤器，您可以在 [Shopify 网站](#) 中找到，以及一组特殊的 3scale 内部过滤器，这些过滤器列在 liquid [参考部分](#) 中列出的。

1.2.2. 上下文

该上下文描述了可在当前页面中使用哪些变量(drops)。基本集包括以下变量：

- provider
- urls
- current_user
- current_account
- today

除了电子邮件模板外，这些变量可在整个开发人员门户的每个页面上提供。但是，大多数内置页面都将具有一些额外的变量。例如，编辑用户表单将公开一个 user 变量（而不是当前用户 - 在编辑用户页面中，用户的身份已经已知）。要检查当前页面中有哪些变量可用，有一个特殊的标签：`{% debug:help %}`。它将添加所有可用的顶级变量的列表，以用作页面源中的注释。

1.2.3. 层次结构

该上下文的直接结果是颠覆性下降被组织到层次结构中。可用的一组变量列出了可用的顶级丢弃。使用它们，您可以更深入地访问层次结构中的元素。例如，如果您想要显示已登录用户的用户名，则需要编写 `{{ current_user.username }}`。

显示较低级别丢弃稍微复杂一些。假定您要为用户显示唯一的应用的名称。查看参考指南，您可以看到方法 `应用` 是帐户丢弃的一部分。这意味着应用是单一应用标签的数组。如果您的用户被允许有多个应用程序，则必须使用逻辑标签迭代应用程序。否则，您可以引用帐户上的第一个（和唯一的）应用程序。要显示的代码如下所示：`{{ current_account.applications.first.name }}`。

1.3. LIQUID 模板

每个 Liquid 模板都包括以下变量：

- [提供者](#) - 您的所有服务、计划和设置
- [urls](#) - 到开发人员门户内置页面的路由 - 登录、注册
- [当前已登录用户的 current_user](#) - 用户名、地址和权限
- [current_account](#) - 消息、应用程序和 [当前已登录用户的计划](#)
- [今天](#) - 当前日期

内置页面也可以有其他可用的变量，这些变量在 Developer Portal 中提到。例如，编辑用户表单编辑将分配或显示应用程序详细信息的 **user** 变量，您可以预期变量 **application** 可以访问。

变量的类型对于使用这个引用非常重要，可以通过将 `{% debug:help %}` 标签放在页面中来为您列出所有可用变量及其类型。但是，从方法或变量名称通常会很容易地猜测它们。

1.4. DROPS

- [Account drop](#)
- [AccountPlan drop](#)
- [Alert drop](#)
- [Application drop](#)
- [ApplicationKey drop](#)
- [ApplicationPlan drop](#)
- [Base drop](#)
- [Base drop](#)
- [BillingAddressField drop](#)
- [Can drop](#)
- [Can drop](#)
- [Cas drop](#)
- [Contract drop](#)
- [Country drop](#)
- [CountryField drop](#)
- [CurrentUser drop](#)
- [Error drop](#)
- [Errors drop](#)
- [Feature drop](#)
- [Field drop](#)
- [Flash drop](#)
- [Forum drop](#)
- [I18n drop](#)
- [Invitation drop](#)
- [Invoice drop](#)

-
- [LineItem drop](#)
 - [Message drop](#)
 - [Message drop](#)
 - [Metric drop](#)
 - [Page drop](#)
 - [Page drop](#)
 - [Pagination drop](#)
 - [part drop](#)
 - [PaymentGateway drop](#)
 - [PaymentTransaction drop](#)
 - [PlanFeature drop](#)
 - [Post drop](#)
 - [Post drop](#)
 - [PricingRule drop](#)
 - [Provider drop](#)
 - [ReferrerFilter drop](#)
 - [Request drop](#)
 - [Role drop](#)
 - [Search drop](#)
 - [Service drop](#)
 - [ServiceContract drop](#)
 - [ServicePlan drop](#)
 - [TimeZone drop](#)
 - [Today drop](#)
 - [Topic drop](#)
 - [Topic drop](#)
 - [URL drop](#)
 - [URLs drop](#)
 - [UsageLimit drop](#)

- [User drop](#)

1.5. TAGS

- [Tag 'braintree_customer_form'](#)
- [Tag 'csrf'](#)
- [Tag 'content'](#)
- [Tag 'content_for'](#)
- [Tag 'debug'](#)
- [Tag 'email'](#)
- [Tag 'flash'](#)
- [Tag 'footer'](#)
- [Tag 'form'](#)
- [Tag 'latest_forum_posts'](#)
- [Tag 'latest_messages'](#)
- [Tag 'logo'](#)
- [Tag 'menu'](#)
- [Tag 'oldfooter'](#)
- [Tag 'plan_widget'](#)
- [Tag 'portlet'](#)
- [Tag 'submenu'](#)
- [Tag '3scale_essentials'](#)
- [Tag 'user_widget'](#)

1.6. 过滤器

- [FormHelpers filters](#)
- [ParamFilter filters](#)
- [Common filters](#)

1.7. DROPS (UP)

1.7.1. Account drop (up)

开发人员帐户.如果要查找电子邮件地址或类似信息, 请参阅 [用户 drop](#)。

```

<h2>Account organization name {{ current_account.name }}</h2>
Plan {{ current_account.bought_account_plan.name }}
Telephone {{ current_account.telephone_number }}

{{ current_account.fields_plain_text }}
{{ current_account.extra_fields_plain_text }}

{% if current_account.approval_required? %}
  <p>This account requires approval.</p>
{% endif %}

{% if current_account.credit_card_required? %}

  {% if current_account.credit_card_stored? %}
    <p>This account has credit card details stored in database.</p>
  {% else %}
    <p>Please enter your {{ 'credit card details' | link_to: urls.payment_details }}.</p>
  {% endif %}

  {% if current_account.credit_card_missing? %}
    <p>This account has no credit card details stored in database.</p>
  {% endif %}
{% endif %}

```

1.7.1.1. 方法

1.7.1.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ account.errors.name | inline_errors }}

```

1.7.1.1.2. id

返回帐户的 id。

1.7.1.1.3. 名称

返回开发人员帐户的组织名称。

1.7.1.1.4. vat_zero_text

返回有关 vat 零的文本。

1.7.1.1.5. vat_rate

返回 vat 速率。

1.7.1.1.6. unread_messages

未读取消息。

1.7.1.1.7. latest_messages

返回最新的消息。

1.7.1.1.8. bought_account_plan

返回帐户已合同的计划。

1.7.1.1.9. bought_account_contract

返回合同帐户。

1.7.1.1.10. credit_card_display_number

1.7.1.1.11. credit_card_expiration_date

1.7.1.1.12. credit_card_required?

返回帐户是否需要输入信用卡详细信息。

1.7.1.1.13. credit_card_stored?

返回帐户是否存储了信用卡详细信息。

1.7.1.1.14. credit_card_missing?

返回帐户没有存储信用卡详情。

1.7.1.1.15. timezone

返回此帐户的时区。

1.7.1.1.16. paid?

返回帐户是否至少具有付费合同。

1.7.1.1.17. on_trial?

返回帐户是否处于试用期，即其所有付费合同都必须处于试用期。

1.7.1.1.18. telephone_number

返回帐户的电话号码。

1.7.1.1.19. approval_required?

返回帐户是否需要审批？

1.7.1.1.20. created_at

返回创建帐户的 UNIX 时间戳(signup)。示例：将时间戳转换为 JavaScript 日期。

```
<script>
  var data = new Date({{ account.created_at }} * 1000);
</script>
```

1.7.1.1.21. full_address

由法律地址、城市和州组成。

1.7.1.1.22. 应用程序

返回帐户的应用。

1.7.1.1.23. subscribed_services

返回具有 ServiceContract drops 的数组。

1.7.1.1.24. admin

返回此帐户的 admin 用户。

1.7.1.1.25. extra_fields_plain_text

返回为帐户定义的额外字段，格式为纯文本。

1.7.1.1.26. fields_plain_text

返回为帐户定义的字段纯文本。

1.7.1.1.27. extra_fields

仅返回带有此帐户值的额外字段。**示例**：打印所有额外字段。

```
{% for field in account.extra_fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.1.1.28. fields

返回所有带有此帐户值的字段。**示例**：打印所有字段。

```
{% for field in account.fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.1.1.29. builtin_fields

1.7.1.1.30. multi_applications_allowed?

1.7.1.1.31. billing_address

返回此帐户的计费地址。

1.7.1.1.32. has_billing_address?

返回此帐户是否具有计费地址。

1.7.1.1.33. can

授予对权限方法的访问权限。

```
{% if account.can.be_deleted? %}  
  <!-- do something -->  
{% endif %}
```

1.7.1.1.34. edit_url

1.7.1.1.35. edit_ogone_billing_address_url

1.7.1.1.36. edit_payment_express_billing_address_url

1.7.1.1.37. edit_braintree_blue_credit_card_details_url

1.7.1.1.38. domain

1.7.1.1.39. upgraded?

1.7.1.1.40. requires_credit_card?

1.7.1.1.41. support_email

1.7.1.1.42. finance_support_email

1.7.2. AccountPlan drop (up)

示例： 使用帐户计划丢弃。

```
<p class="notice">The examples for plan drop apply here</p>
```

1.7.2.1. Methods

1.7.2.1.1. selected?

返回是否选择了计划。

```
{% if plan.selected? %}  
  <p>You will signup to {{ plan.name }}</p>  
{% endif %}
```

1.7.2.1.2. bought?

返回计划是否购买。

```
{% if plan.bought? %}  
  <p>You are on this plan already!</p>  
{% endif %}
```


1.7.2.1.3. 功能

返回一组可用功能。

1.7.2.1.4. setup_fee

返回设置费用。

1.7.2.1.5. name

返回计划的名称。

```
<h2>We offer you a new {{ plan.name }} plan!</h2>
```

1.7.2.1.6. system_name

返回计划的系统名称。

```
{% for plan in available_plans %}
  {% if plan.system_name == 'my_free_plan' %}

    <p>You will buy our only free plan!</p>
  {% endif %}
{% endfor %}
```

1.7.2.1.7. id

返回计划 ID。

1.7.2.1.8. free?

如果计划不是 **paid**，则为免费计划。请参阅 *paid?* method.

```
{% if plan.free? %}
  <p>This plan is free of charge.</p>
{% else %}

  <p>Plan costs</p>
  Setup fee {{ plan.setup_fee }}
  Flat cost {{ plan.flat_cost }}

{% endif %}
```

1.7.2.1.9. paid?

当计划具有非零的固定或设置费用或存在一些定价规则时，计划为 **paid**。

```
{% if plan.paid? %}
  <p>this plan is a paid one.</p>
{% else %}
  <p>this plan is a free one.</p>
{% endif %}
```

1.7.2.1.10. approval_required?

返回计划是否需要审批？

```
{% if plan.approval_required? %}
  <p>This plan requires approval.</p>
{% endif %}
```

1.7.2.1.11. flat_cost

返回计划的月度固定费用。

1.7.3. Alert drop (up)

示例：在 liquid 中使用警报丢弃。

```
<h2>Alert details</h2>
Level {{ alert.level }}
Message {{ alert.message }}
Utilization {{ alert.utilization }}
```

1.7.3.1. Methods

1.7.3.1.1. level

警报级别可以是 50, 80, 90, 100, 120, 150, 200, 300。

1.7.3.1.2. message

描述警报的文本消息，例如 *hits per minute: 5 of 5*。

1.7.3.1.3. 使用率

标记触发警报 1.0 的实际利用率的十进制数等于 100%。

```
Used by {{ alert.utilization | times: 100 }} percent.
```

1.7.4. Application drop (up)

示例：在 liquid 中使用应用程序丢弃。

```
<h2>Application {{ application.name }} ({{ application.application_id }})</h2>
<p>{{ application.description }}</p>
```

1.7.4.1. Methods

1.7.4.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ application.errors.name | inline_errors }}
```

-

1.7.4.1.2. id

返回应用程序的 id。

1.7.4.1.3. can_change_plan?

如果允许直接或通过请求更改应用，则返回 **true**。

1.7.4.1.4. trial?

如果合同仍在试用期间，返回真。

备注：如果您更改计划的试用期限，则不影响现有合同。

1.7.4.1.5. live?

1.7.4.1.6. state

有三种可能的状态：

- pending
- live
- suspended

1.7.4.1.7. remaining_trial_period_days

试用期间仍存在的天数。

1.7.4.1.8. plan

使用应用计划返回计划 drop。

1.7.4.1.9. plan_change_permission_name

返回允许操作的名称

1.7.4.1.10. plan_change_permission_warning

返回允许操作的警告消息。

1.7.4.1.11. 合同

1.7.4.1.12. admin_url

返回应用的 admin_url。

1.7.4.1.13. name

返回应用程序的名称。

1.7.4.1.14. can

1.7.4.1.15. oauth

1.7.4.1.16. 待定？

如果应用程序状态为 pending，则返回 **true**。

1.7.4.1.17. buyer_alerts_enabled?

1.7.4.1.18. 描述

返回应用程序的描述。

1.7.4.1.19. redirect_url

返回应用 OAuth 的重定向 URL。

1.7.4.1.20. filters_limit

返回此应用程序允许的引用器过滤器量。

1.7.4.1.21. keys_limit

返回此应用程序允许的应用程序密钥量。

1.7.4.1.22. referrer_filters

返回与此应用程序关联的引用器过滤器。

1.7.4.1.23. rejection_reason

返回拒绝应用程序的原因。

1.7.4.1.24. user_key

返回应用程序的 user_key。

1.7.4.1.25. application_id

返回应用程序的 application_id。

1.7.4.1.26. key

返回应用程序 ID 或用户密钥。

1.7.4.1.27. url

返回此应用的内置详情视图的 URL。

1.7.4.1.28. edit_url

返回此应用的内置编辑视图的 URL。

1.7.4.1.29. update_user_key_url

1.7.4.1.30. log_requests_url

1.7.4.1.31. alerts_url

1.7.4.1.32. application_keys_url

1.7.4.1.33. service

该应用所属的服务。

1.7.4.1.34. keys

返回应用的密钥。

```
{% case application.keys.size %}
{% when 0 %}
  Generate your application key.
{% when 1 %}
  <h4>Application key for {{ application.name }} {{ application.application_id }}</h4>
  <p>Key is: {{ application.keys.first }}</p>
{% else %}
  <h4>Application keys for {{ application.name }} {{ application.application_id }}</h4>
  <ul>
    {% for key in application.keys %}
      <li>{{ key }}</li>
    {% endfor %}
  </ul>
{% endcase %}
```

1.7.4.1.35. user_key_mode?

1.7.4.1.36. app_id_mode?

1.7.4.1.37. change_plan_url

1.7.4.1.38. log_requests?

1.7.4.1.39. application_keys

1.7.4.1.40. extra_fields

返回非隐藏的额外字段以及此应用程序的值。**示例**：打印所有额外字段。

```
{% for field in application.extra_fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.4.1.41. fields

返回所有内置字段和额外字段以及此应用的值。**示例**：打印所有字段。

```
{% for field in application.fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.4.1.42. builtin_fields

仅返回应用的内置字段。

1.7.4.1.43. cinstance

1.7.5. ApplicationKey drop (up)

1.7.5.1. 方法

1.7.5.1.1. id

1.7.5.1.2. 值

1.7.5.1.3. url

1.7.5.1.4. 应用

1.7.6. ApplicationPlan drop (up)

1.7.6.1. 方法

1.7.6.1.1. selected?

返回是否选择了计划。

```
{% if plan.selected? %}
  <p>You will signup to {{ plan.name }}</p>
{% endif %}
```

1.7.6.1.2. bought?

返回计划是否购买。

```
{% if plan.bought? %}
  <p>You are on this plan already!</p>
{% endif %}
```

1.7.6.1.3. 功能

返回计划可见的功能。

```
{% if plan == my_free_plan %}
  <p>These plans are the same.</p>
{% else %}
```

```
<p>These plans are not the same.</p>
{% endif %}
```

1.7.6.1.4. setup_fee

返回计划的设置费用。

1.7.6.1.5. name

返回计划的名称。

```
<h2>We offer you a new {{ plan.name }} plan!</h2>
```

1.7.6.1.6. system_name

返回计划的系统名称。

```
{% for plan in available_plans %}
  {% if plan.system_name == 'my_free_plan' %}

    <p>You will buy our only free plan!</p>
  {% endif %}
{% endfor %}
```

1.7.6.1.7. id

返回计划 ID。

1.7.6.1.8. free?

如果计划不是 **paid**，则为免费计划。请参阅 *paid?* method.

```
{% if plan.free? %}
  <p>This plan is free of charge.</p>
{% else %}

  <p>Plan costs</p>
  Setup fee {{ plan.setup_fee }}
  Flat cost {{ plan.flat_cost }}

{% endif %}
```

1.7.6.1.9. paid?

当计划具有非零的固定或设置费用或存在一些定价规则时，计划为 **paid**。

```
{% if plan.paid? %}
  <p>this plan is a paid one.</p>
{% else %}
  <p>this plan is a free one.</p>
{% endif %}
```

1.7.6.1.10. approval_required?

返回计划是否需要审批？

```
{% if plan.approval_required? %}  
  <p>This plan requires approval.</p>  
{% endif %}
```

1.7.6.1.11. flat_cost

返回计划的月度固定费用。

1.7.6.1.12. metrics

返回计划的指标。

1.7.6.1.13. usage_limits

返回计划用量限制。

1.7.6.1.14. service

返回计划服务。

1.7.7. Base drop (up)

1.7.7.1. 方法

1.7.7.1.1. login_url

1.7.7.1.2. user_identified?

1.7.8. Base drop (up)

1.7.8.1. 方法

1.7.8.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ base.errors.name | inline_errors }}
```

1.7.8.1.2. title

返回标题结果。

1.7.8.1.3. kind

返回结果类型，可以是 'topic' 或 'page'。

1.7.8.1.4. url

返回结果的资源 URL。

1.7.8.1.5. 描述

为结果返回一个描述性字符串。

1.7.9. BillingAddressField drop (up)

1.7.9.1. 方法

1.7.9.1.1. input_name

1.7.9.1.2. label

1.7.9.1.3. choices

1.7.9.1.4. 错误

1.7.9.1.5. html_id

1.7.9.1.6. hidden?

1.7.9.1.7. visible?

1.7.9.1.8. read_only?

1.7.9.1.9. name

1.7.9.1.10. value

1.7.9.1.11. required

1.7.10. Can drop (up)

1.7.10.1. 方法

1.7.10.1.1. be_updated?

1.7.10.1.2. add_referrer_filters?

1.7.10.1.3. add_application_keys?

1.7.10.1.4. regenerate_user_key?

1.7.10.1.5. regenerate_oauth_secret?

1.7.10.1.6. manage_keys?

1.7.10.1.7. delete_key?

1.7.11. Can drop (up)

1.7.11.1. 方法

1.7.11.1.1. change_plan?

1.7.12. Cas drop (up)

1.7.12.1. 方法

1.7.12.1.1. login_url

1.7.12.1.2. user_identified?

1.7.13. Contract drop (up)

Plan of the contract {{ contract.plan.name }}

1.7.13.1. 方法

1.7.13.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

{{ contract.errors.name | inline_errors }}

1.7.13.1.2. id

返回 id

1.7.13.1.3. can_change_plan?

如果可能采用任何形式的更改，则返回 true。

1.7.13.1.4. trial?

如果合同仍在试用期间，返回真。



注意

如果您更改计划的试用期限，则不影响现有合同。

1.7.13.1.5. live?

1.7.13.1.6. state

有三种可能的状态：

- pending
- live
- suspended

1.7.13.1.7. remaining_trial_period_days

试用期间仍存在的天数。

1.7.13.1.8. plan

返回合同计划。

1.7.13.1.9. plan_change_permission_name

返回允许操作的名称

1.7.13.1.10. plan_change_permission_warning

返回允许操作的警告消息。

1.7.13.1.11. 合同

1.7.14. Country drop (up)

1.7.14.1. 方法

1.7.14.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ country.errors.name | inline_errors }}
```

1.7.14.1.2. to_str

1.7.14.1.3. code

1.7.14.1.4. label

1.7.15. CountryField drop (up)

1.7.15.1. 方法

1.7.15.1.1. value

返回国家 ID。

```
{{ account.fields.country.value }} => 42
```

compare with:

```
{{ account.fields.country }} => 'United States'
```

1.7.15.1.2. name

返回字段的系统名称。

1.7.15.1.3. required

1.7.15.1.4. hidden?

1.7.15.1.5. hidden

1.7.15.1.6. visible?

1.7.15.1.7. visible

1.7.15.1.8. read_only

1.7.15.1.9. 错误

1.7.15.1.10. input_name

1.7.15.1.11. html_id

1.7.15.1.12. label

返回字段的标签。

```
{{ account.fields.country.label }}  
<!-- => 'Country' -->
```

1.7.15.1.13. to_str

返回国家名称。

```
{{ account.fields.country }} => 'United States'
```

1.7.15.1.14. choices

1.7.16. CurrentUser drop (up)

1.7.16.1. 方法

1.7.16.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ current_user.errors.name | inline_errors }}
```

1.7.16.1.2. admin?

返回用户是否为 admin。

```
{% if user.admin? %}
  <p>You are an admin of your account.</p>
{% endif %}
```

1.7.16.1.3. username

返回用户的用户名（html 转意）。

1.7.16.1.4. account

返回用户的帐户。

1.7.16.1.5. name

返回用户的名和姓。

1.7.16.1.6. email

返回用户的电子邮件。

1.7.16.1.7. password_required?

对于使用内置的 Developer Portal 认证机制的用户返回 **true**，对于经过 Janrain, CAS 或其他单点登陆方法的用户返回 **false**。

```
{{ if user.password_required? }}

{{ endif }}
```

1.7.16.1.8. 部分

返回用户有权访问的部分列表。

```
{% if user.sections.size > 0 %}
  <p>You can access following sections of our portal:</p>
  <ul>
    {% for section in user.sections %}
      <li>{{ section }}</li>
    {% endfor %}
  </ul>
{% endif %}
```

1.7.16.1.9. role

返回用户的角色。

1.7.16.1.10. roles_collection

重新调整用户的可用角色列表

```
{% for role in user.roles_collection %}
  <li>
    <label for="user_role_{{ role.key }}">

      {{ role.text&nbsp;nbsp; }}
    </label>
  </li>
{% endfor %}
```

1.7.16.1.11. url

返回用户的资源 URL。

```
{{ 'Delete' | delete_button: user.url }}
```

1.7.16.1.12. edit_url

返回 URL 以编辑用户。

```
{{ 'Edit' | link_to: user.edit_url, title: 'Edit', class: 'action edit' }}
```

1.7.16.1.13. can

公开当前用户的权限，这取决于您的设置和用户角色。您可以在返回的对象上调用这些方法：

- requests_user?
- create_application?
- see_log_requests?

```
{% if current_user.can.see_log_requests? and application.log_requests? %}
  (<a href="{{ application.log_requests_url }}" class="action edit">App Request Log</a>)
{% endif %}
```

1.7.16.1.14. extra_fields

返回带有此用户值的非隐藏的额外字段。**示例**：打印所有额外字段。

```
{% for field in user.extra_fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.16.1.15. fields

返回所有带有此用户值的字段。**示例**：打印所有字段。

```
{% for field in user.fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.16.1.16. builtin_fields

返回所有带有此用户值的内置字段。

1.7.17. Error drop (up)

当表单因为无效数据而无法提交时，错误 数组将位于相关模型中。

1.7.17.1. 方法

1.7.17.1.1. attribute

模型的返回属性到这个错误是相关的。

```
{{ account.errors.org_name.first.attribute }}
<!-- org_name -->
```

1.7.17.1.2. message

返回错误的描述。

```
{{ account.errors.first.message }}
<!-- cannot be blank -->
```

1.7.17.1.3. value

返回与 错误 相关的属性值。

```
{{ account.errors.org_name.first.value }}
<!-- ==> "ACME Co." -->
```

1.7.17.1.4. to_str

返回错误的完整描述（包括属性名称）。

```
{{ model.errors.first }}
<!-- ==> "Attribute can't be blank" -->
```

1.7.18. Errors drop (up)

示例： 获取所有错误。

```
{% for error in form.errors %}
  attribute: {{ error.attribute }}
  ...
{% endfor %}
```

1.7.18.1. 方法

1.7.18.1.1. empty?

如果没有错误，返回 true。

```
{% if form.errors == empty %}
  Congratulations! You have no errors!
{% endif %}
```

1.7.18.1.2. present?

如果有错误，返回 true。

```
{% if form.errors == present %}
  Sorry, there were some errors.
{% endif %}
```

1.7.19. Feature drop (up)

1.7.19.1. 方法

1.7.19.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ feature.errors.name | inline_errors }}
```

1.7.19.1.2. name

返回功能的名称。

```
<h2>Feature {{ feature.name }}</h2>
```

1.7.19.1.3. description

返回该功能的描述。

1.7.19.1.4. has_description?

返回功能是否有描述。

```
{% if feature.has_description? %}
  {{ feature.description }}
{% else %}
  This feature has no description.
{% endif %}
```

1.7.20. Field drop (up)

1.7.20.1. 方法

1.7.20.1.1. value

返回字段的值。

```
Name: {{ account.fields.first_name.value }}
```

1.7.20.1.2. name

返回字段的系统名称。

1.7.20.1.3. required

1.7.20.1.4. hidden?

1.7.20.1.5. hidden

1.7.20.1.6. visible?

1.7.20.1.7. visible

1.7.20.1.8. read_only

1.7.20.1.9. 错误

1.7.20.1.10. input_name

返回提交表单时预期的 HTML 输入的名称。

```
<!-- the 'name' attribute will be 'account[country]' -->
```

1.7.20.1.11. html_id

返回唯一字段标识符，该标识符通常用作 HTML ID 属性。

```
{{ account.fields.country.html_id }}  
<!-- => 'account_country' -->
```

1.7.20.1.12. label

返回字段的标签。

```
{{ account.fields.country.label }}  
<!-- => 'Country' -->
```

1.7.20.1.13. to_str

如果用作变量，则返回字段的值。

```
{{ account.fields.first_name }} => 'Tom'
```

1.7.20.1.14. choices

返回可用于该字段的一组选择（若有）。例如，对于名为 **fruit** 的字段，它可以响应 ['apple', 'bannana', 'orange']。

您可以在 3scale 管理门户的 **Audience > Accounts > Settings > Fields Definitions** 下定义选择。每个数组元素都响应通常只是相同的 **id** 和 **label**，除非字段是特殊的内置项（如 **country**）。建议您使用这些方法而不是输出选择 "as is" 以备将来的兼容性。

```
{% for choice in field.choices %}
  <select name="{{ field.input_name }}" id="{{ field.html_id }}_id"
    class="{{ field.errors | error_class }}">
    <option {% if field.value == choice %} selected {% endif %} value="{{ choice.id }}">
      {{ choice }}
    </option>
{% endfor %}
```

1.7.21. Flash drop (up)

1.7.21.1. 方法

1.7.21.1.1. messages

返回一组信息。

```
{% for message in flash.messages %}
  <p id="flash-{{ message.type }}">
    {{ message.text }}
  </p>
{% endfor %}
```

1.7.22. Forum drop (up)

1.7.22.1. 方法

1.7.22.1.1. enabled?

如果您启用了论坛功能，则返回 **true**。

```
{% if forum.enabled? %}
  <a href="/forum">Check out our forum!</a>
{% endif %}
```

1.7.22.1.2. latest_posts

1.7.23. I18n drop (up)

为 i18n 支持提供有用的字符串。

```

{{ object.some_date | date: i18n.long_date }}

```

1.7.23.1. 方法

1.7.23.1.1. short_date

%b %d 的别名

```

Dec 11

```

1.7.23.1.2. long_date

%B %d, %Y 的别名

```

December 11, 2013

```

1.7.23.1.3. default_date

%Y-%m-%d 的别名

```

2013-12-11

```

1.7.24. Invitation drop (up)

Email: {{ invitation.email }}

```

<tr id="invitation_{{ invitation.id }}">
  <td> {{ invitation.email }} </td>
  <td> {{ invitation.sent_at | date: i18n.short_date }} </td>
  <td>
    {% if invitation.accepted? %}
      yes, on {{invitation.accepted_at | format: i18n.short_date }}
    {% else %}
      no
    {% endif %}
  </td>
</tr>

```

1.7.24.1. 方法

1.7.24.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ invitation.errors.name | inline_errors }}

```

1.7.24.1.2. email

返回电子邮件地址。

1.7.24.1.3. accepted?

如果邀请被接受，则返回 true。

1.7.24.1.4. accepted_at

如果邀请被接受，则返回日期。

```
{{ invitation.accepted_at | date: i18n.short_date }}
```

1.7.24.1.5. sent_at

返回创建日期。

```
{{ invitation.sent_at | date: i18n.short_date }}
```

1.7.24.1.6. resend_url

返回重新发送邀请的 URL。

```
{{ "Resend" | update_button: invitation.resend_url }}
```

1.7.24.1.7. url

返回资源 URL。

```
{{ "Delete" | delete_button: invitation.url }}
```

1.7.25. Invoice drop (up)

1.7.25.1. 方法

1.7.25.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ invoice.errors.name | inline_errors }}
```

1.7.25.1.2. friendly_id

返回友好 ID。

```
<td> {{ invoice.id }} </td>  
<td> {{ invoice.name }} </td>  
<td> {{ invoice.state }} </td>  
<td> {{ invoice.cost }} {{ invoice.currency }} </td>
```

1.7.25.1.3. name

由月份和年份组成的字符串。

1.7.25.1.4. state

1.7.25.1.5. cost

返回带有两个十进制的数字。

```
23.00
```

1.7.25.1.6. currency

1.7.25.1.7. cost_without_vat

返回没有 VAT 的成本。

1.7.25.1.8. vat_amount

返回 vat 数量。

1.7.25.1.9. exists_pdf?

如果生成 PDF, 则返回 true。

1.7.25.1.10. period_begin

```
{{ invoice.period_begin | date: i18n.short_date }}
```

1.7.25.1.11. period_end

```
{{ invoice.period_end | date: i18n.long_date }}
```

1.7.25.1.12. issued_on

```
{{ invoice.issued_on | date: i18n.long_date }}
```

1.7.25.1.13. due_on

```
{{ invoice.due_on | date: i18n.long_date }}
```

1.7.25.1.14. paid_on

```
{{ invoice.paid_on | date: i18n.long_date }}
```

1.7.25.1.15. vat_code

1.7.25.1.16. fiscal_code

1.7.25.1.17. account

返回 AccountDrop。

1.7.25.1.18. buyer_account

1.7.25.1.19. line_items

返回 LineItemDrop 数组。

```
{% for line_item in invoice.line_items %}
  <tr class="line_item {% cycle 'odd', 'even' %}">
    <th>{{ line_item.name }}</th>
    <td>{{ line_item.description }}</td>
    <td>{{ line_item.quantity }}</td>
    <td>{{ line_item.cost }}</td>
  </tr>
{% endfor %}
```

1.7.25.1.20. payment_transactions

返回 PaymentTransactionDrop 数组。

```
{% for payment_transaction in invoice.payment_transactions %}
  <tr>
    <td> {% if payment_transaction.success? %} Success {% else %} Failure {% endif %} </td>
    <td> {{ payment_transaction.created_at }} </td>
    <td> {{ payment_transaction.reference }} </td>
    <td> {{ payment_transaction.message }} </td>
    <td> {{ payment_transaction.amount }} {{ payment_transaction.currency }} </td>
  </tr>
{% endfor %}
```

1.7.25.1.21. url

返回发票的资源 URL。

```
{{ "Show" | link_to: invoice.url }}
```

1.7.25.1.22. pdf_url

返回发票 PDF 的资源 URL。

```
{{ "PDF" | link_to: invoice.pdf_url }}
```

1.7.26. Janrain drop (up)

1.7.26.1. 方法

1.7.26.1.1. login_url

1.7.26.1.2. user_identified?

1.7.26.1.3. session_url

1.7.26.1.4. relying_party

1.7.27. Lineltem drop (up)

1.7.27.1. 方法

1.7.27.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ line_item.errors.name | inline_errors }}
```

1.7.27.1.2. name

```
{% for line_item in invoice.line_items %}
  <tr class="line_item {% cycle 'odd', 'even' %}">
    <th>{{ line_item.name }}</th>
    <td>{{ line_item.description }}</td>
    <td>{{ line_item.quantity }}</td>
    <td>{{ line_item.cost }}</td>
  </tr>
{% endfor %}
```

1.7.27.1.3. description

1.7.27.1.4. quantity

1.7.27.1.5. cost

1.7.28. Message drop (up)

1.7.28.1. 方法

1.7.28.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ message.errors.name | inline_errors }}
```

1.7.28.1.2. id

返回消息的 id。

1.7.28.1.3. 主题

如果主题不存在，则返回截断正文或 **(无主题)** 字符串。

1.7.28.1.4. 正文 (body)

邮件正文。

1.7.28.1.5. created_at

返回创建日期。

```
{{ message.created_at | date: i18n.short_date }}
```

1.7.28.1.6. url

消息详细信息的 URL，指向 inbox 或 outbox。

1.7.28.1.7. state

'read' 或 'unread'。

1.7.28.1.8. sender

返回发送者的名称。

1.7.28.1.9. 至

返回接收方的名称。

1.7.28.1.10. 接收者

1.7.29. Message drop (up)

1.7.29.1. 方法

1.7.29.1.1. type

可能的信息类型有：

- 成功（目前尚未使用）
- info
- warning
- danger

1.7.29.1.2. text

1.7.30. Metric drop (up)

1.7.30.1. 方法

1.7.30.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ metric.errors.name | inline_errors }}
```

1.7.30.1.2. unit

返回指标的单元。

```
This metric is measured in {{ metric.unit | pluralize }}
```

1.7.30.1.3. description

返回指标的描述。

1.7.30.1.4. name

返回指标的名称。

```
<h4>Metric {{ metric.name }}</h4>
<p>{{ metric.description }}</p>
```

1.7.30.1.5. system_name

返回此指标的系统名称。

```
<h4>Metric {{ metric.name }}</h4>
<p>{{ metric.system_name }}</p>
```

1.7.30.1.6. usage_limits

返回指标用量限制。

```
{% if metric.usage_limits.size > 0 %}
  <p>Usage limits of the metric</p>
  <ul>
    {% for usage_limit in metric.usage_limits %}
      <li>{{ usage_limit.period }} : {{ usage_limit.value }}</li>
    {% endfor %}
  </ul>
{% else %}
  <p>This metric has no usage limits</p>
{% endif %}
```

1.7.30.1.7. pricing_rules

返回指标的定价规则

```
{% if metric.pricing_rules.size > 0 %}
  <p>Pricing rules of the metric</p>
  <ul>
    {% for pricing_rule in metric.pricing_rules %}
```

```

    <li>{{ pricing_rule.cost_per_unit }}</li>
  {% endfor %}
</ul>

{% else %}
  <p>This metric has no pricing rules</p>
{% endif %}

```

1.7.30.1.8. has_parent

1.7.31. Page drop (up)

1.7.31.1. 方法

1.7.31.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ page.errors.name | inline_errors }}

```

1.7.31.1.2. title

返回页面的标题。

```

<title>{{ page.title }}</title>

```

1.7.31.1.3. system_name

返回页面的系统名称

```

{% if page.system_name == 'my_page' %}
  {% include 'custom_header' %}
{% endif %}

```

1.7.32. Page drop (up)

1.7.32.1. 方法

1.7.32.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ page.errors.name | inline_errors }}

```

1.7.32.1.2. title

1.7.32.1.3. kind

1.7.32.1.4. url

1.7.32.1.5. description

1.7.33. Pagination drop (up)

1.7.33.1. 方法

1.7.33.1.1. page_size

一个整页中的项目数。

```
{% for part in pagination.parts %}
  {% if part.is_link %}
    {% case part.rel %}
      {% when 'previous' %}
        {% assign css_class = 'previous_page' %}
      {% when 'next' %}
        {% assign css_class = 'next_page' %}
      {% else %}
        {% assign css_class = "" %}
      {% endcase %}

    <a class="{{ css_class }}" rel="{{ part.rel }}" href="{{ part.url }}">{{ part.title }}</a>
  {% else %}
    {% case part.rel %}
      {% when 'current' %}
        <em class="current">{{ part.title }}</em>
      {% when 'gap' %}
        &#x2026;
      {% else %}
        {{ part.title }}
      {% endcase %}
    {% endif %}
  {% endfor %}
```

<!-- Outputs:

```
=====

<a class="previous_page" rel="prev" href="?page=7">&#x2190; Previous</a>
<a rel="start" href="?page=1">1</a>
<a href="?page=2">2</a>
<a href="?page=3">3</a>
<a href="?page=4">4</a>
<a href="?page=5">5</a>
<a href="?page=6">6</a>
<a rel="prev" href="?page=7">7</a>
<em class="current">8</em>
<a rel="next" href="?page=9">9</a>
<a href="?page=10">10</a>
<a href="?page=11">11</a>
<a href="?page=12">12</a>
&#x2026;
<a href="?page=267">267</a>
<a href="?page=268">268</a>
<a class="next_page" rel="next" href="?page=9">Next &#x2192;</a>
```

```
=====
-->
```

1.7.33.1.2. `current_page`

当前选定页面的编号。

1.7.33.1.3. `current_offset`

项目到目前为止跳过。

1.7.33.1.4. `pages`

页面总数。

1.7.33.1.5. `items`

所有页面中项目总数。

1.7.33.1.6. `previous`

上一页或空页面的数量。

1.7.33.1.7. `next`

下一页或空页的数量。

1.7.33.1.8. `parts`

有助于呈现用户友好的分页的元素。如需更多信息，请参阅 `[[part-drop[part drop]]`。

1.7.34. `Part drop (up)`

1.7.34.1. 方法

1.7.34.1.1. `url`

1.7.34.1.2. `rel`

1.7.34.1.3. `current?`

1.7.34.1.4. `is_link`

1.7.34.1.5. `title`

1.7.34.1.6. `to_s`

1.7.35. `PaymentGateway drop (up)`

1.7.35.1. Methods

1.7.35.1.1. braintree_blue?

返回当前支付网关是否为 braintreepayments.com。

1.7.35.1.2. authorize_net?

返回当前支付网关是否为 authorize.Net。

1.7.35.1.3. type

返回此支付网关的类型。

1.7.36. PaymentTransaction drop (up)

1.7.36.1. Methods

1.7.36.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ payment_transaction.errors.name | inline_errors }}

```

1.7.36.1.2. currency

返回货币。

```

{% for payment_transaction in invoice.payment_transactions %}
  <tr>
    <td> {% if payment_transaction.success? %} Success {% else %} Failure {% endif %} </td>
    <td> {{ payment_transaction.created_at }} </td>
    <td> {{ payment_transaction.reference }} </td>
    <td> {{ payment_transaction.message }} </td>
    <td> {{ payment_transaction.amount }} {{ payment_transaction.currency }} </td>
  </tr>
{% endfor %}

```

1.7.36.1.3. amount

返回数量。

1.7.36.1.4. created_at

返回创建日期。

1.7.36.1.5. success?

如果成功，返回 true。

1.7.36.1.6. message

返回事务的消息。

1.7.36.1.7. reference

返回引用。

1.7.37. PlanFeature drop (up)

1.7.37.1. Methods

1.7.37.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ plan_feature.errors.name | inline_errors }}
```

1.7.37.1.2. name

返回功能的名称。

```
<h2>Feature {{ feature.name }}</h2>
```

1.7.37.1.3. description

返回该功能的描述。

1.7.37.1.4. has_description?

返回功能是否有描述。

```
{% if feature.has_description? %}
  {{ feature.description }}
{% else %}
  This feature has no description.
{% endif %}
```

1.7.37.1.5. enabled?

1.7.38. Post drop (up)

1.7.38.1. Methods

1.7.38.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ post.errors.name | inline_errors }}
```

1.7.38.1.2. 正文 (body)

post 的文本。

1.7.38.1.3. topic

属于一个 `[[topic-drop[topic]]` 的每个 post

1.7.38.1.4. created_at

此后创建的日期。

```

{{ post.created_at | date: i18n.short_date }}

```

1.7.38.1.5. url

此后期的 URL 位于其主题中。

1.7.39. Post drop (up)

1.7.39.1. Methods

1.7.39.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ post.errors.name | inline_errors }}

```

1.7.39.1.2. title

1.7.39.1.3. kind

1.7.39.1.4. url

1.7.39.1.5. description

1.7.40. PricingRule drop (up)

1.7.40.1. Methods

1.7.40.1.1. cost_per_unit

返回定价规则的单位成本。**示例**：在 liquid 中使用定价规则丢弃。

```

<h2>Pricing rule</h2>
Min value {{ pricing_rule.min }}
Max value {{ pricing_rule.max }}
Cost per unit {{ pricing_rule.cost_per_unit }}

```

1.7.40.1.2. 分钟

返回定价规则的最小值。

1.7.40.1.3. max

返回定价规则的最大值。

1.7.40.1.4. plan

返回定价规则计划。

1.7.41. Provider drop (up)

1.7.41.1. Methods

1.7.41.1.1. name

返回您的组织名称。

```
Domain {{ provider.domain }}

{% if provider.multiple_applications_allowed? %}

  <p>Applications</p>
  <ul>
    {% for app in account.applications %}
      <li>{{ app.name }}</li>
    {% endfor %}
  </ul>

{% else %}
  Application {{ account.applications.first.name }}
{% endif %}

For general questions, contact us at {{ provider.support_email }},
for invoice or payment related questions, contact us at {{ provider.finance_support_email }}
```

1.7.41.1.2. payment_gateway

返回与您的组织关联的支付网关。

1.7.41.1.3. domain

开发人员门户的域。

1.7.41.1.4. timezone

返回您使用的时区。您可以在管理门户中更改您的时区。选择 **Dashboard > Account Settings**。在 **Overview** 页面中，向下滚动到 **Account Details**，单击 **Edit** 并修改 **Time Zone** 字段。

1.7.41.1.5. support_email

帐户的支持电子邮件。

1.7.41.1.6. finance_support_email

帐户的财务支持电子邮件。

1.7.41.1.7. telephone_number

返回帐户的电话号码。

1.7.41.1.8. multi_applications_allowed?

如果开发人员可以使用自己的密钥、stats 等更多独立的应用程序，则为 true。这取决于您的 3scale 计划。

```
{% if provider.multiple_applications_allowed? %}

  <p>Applications</p>
  <ul>
    {% for app in account.applications %}
      <li>{{ app.name }}</li>
    {% endfor %}
  </ul>

{% else %}
  Application {{ account.applications.first.name }}
{% endif %}
```

1.7.41.1.9. logo_url

返回徽标 URL

1.7.41.1.10. multiple_services_allowed?

如果您的 3scale 计划允许您将多个 API 作为单独的服务进行管理，则为 True。

```
{% if provider.multiple_services_allowed? %}
  {% for service in provider.services %}
    Service {{ service.name }} is available.
  {% endfor %}
{% endif %}
```

1.7.41.1.11. finance_allowed?

1.7.41.1.12. multi_users_allowed?

如果开发人员帐户可以关联多次登录则为 true。这取决于您的 3scale 计划，且在管理门户的管理门户中是否在 **Audience > Developer Portal > Feature Visibility** 下打开了其可见性。

```
{% if provider.multiple_users_allowed? %}
  <ul id="subsubmenu">
    <li>
      {{ 'Users' | link_to: urls.users }}
    </li>
  </ul>
```

```

</li>
  {{ 'Sent invitations' | link_to: urls.invitations }}
</li>
</ul>
{% endif %}

```

1.7.41.1.13. account_plans

返回所有公布的帐户计划。

```

<p>We offer following account plans:</p>
<ul>
  {% for plan in model.account_plans %}
    <li>{{ plan.name }} </li>
  {% endfor %}
</ul>

```

1.7.41.1.14. services

返回所有定义的服务。

```

<p>You can signup to any of our services!</p>
<ul>
  {% for service in provider.services %}
    <li>{{ service.name }} <a href="/signup/service/{{ service.system_name }}">Signup!</a></li>
  {% endfor %}

```

1.7.41.1.15. signups_enabled?

您可以在 3scale 管理门户的 **Audience > Accounts > Settings > Usage Rules** 下启用或禁用注册。

1.7.41.1.16. account_management_enabled?

您可以在 3scale 管理门户的 **Audience > Accounts > Settings > Usage Rules** 下启用或禁用帐户管理

1.7.42. ReferrerFilter drop (up)

1.7.42.1. Methods

1.7.42.1.1. id

1.7.42.1.2. value

1.7.42.1.3. delete_url

1.7.42.1.4. application

1.7.43. Request drop (up)

示例：在 liquid 中使用请求丢弃。

-

```
<h2>Request details</h2>
URI {{ request.request_uri }}
Host {{ request.host }}
Host and port {{ request.host_with_port }}
```

1.7.43.1. Methods

1.7.43.1.1. request_uri

返回请求的 URI。

1.7.43.1.2. host_with_port

使用请求的端口返回主机。

1.7.43.1.3. 主机

返回请求 URL 的主机部分。

1.7.43.1.4. path

返回请求 URL 的路径部分。

```
{% if request.path == '/' %}
  Welcome on a landing page!
{% else %}
  This just an ordinary page.
{% endif %}
```

1.7.44. Role drop (up)

1.7.44.1. Methods

1.7.44.1.1. name

返回角色的内部名称，对系统很重要。

1.7.44.1.2. description

返回角色的描述性文本。

1.7.45. Search drop (up)

1.7.45.1. Methods

1.7.45.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ search.errors.name | inline_errors }}
```

1.7.45.1.2. query

返回搜索的字符串

```

<h3>{{ search.token }}</h3>
<p>found on {{ search.total_found }} {{ search.item | pluralize }} </p>
<dl>
  {% for result in search.results %}
    <dt>
      [ {{ result.kind | capitalize}} ]
      {{ result.title | link_to: result.url }}
    </dt>
    <dd>
      {{ result.description }}
    </dd>
  {% endfor %}
</dl>

```

1.7.45.1.3. total_found

返回匹配元素的数量。

1.7.45.1.4. results

返回队列搜索的结果数组。

1.7.46. Service drop (up)

1.7.46.1. Methods

1.7.46.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```

{{ service.errors.name | inline_errors }}

```

1.7.46.1.2. name

返回服务名称。

1.7.46.1.3. system_name

返回服务的系统名称。

```

{% case service.system_name %}
{% when 'api' %}
  API is our newest service!
{% when 'old' %}
  Unfortunately, we do not allow more signups to our old service.
{% endcase %}

```

1.7.46.1.4. description

返回服务的描述。

1.7.46.1.5. subscribed?

返回服务是否已订阅。

```
{% if service.subscribed? %}
  <p>You already subscribed this service.</p>
{% endif %}
```

1.7.46.1.6. subscription

如果当前登录的用户订阅此服务，则返回订阅(**ServiceContract drop**)。

```
{% if service.subscription %}
  Your applications for service {{ service.name }} are:
  {% for app in service.subscription.applications %}
    {{ app.name }}<br/>
  {% endfor %}
{% else %}
  <p>You are not subscribed to this.</p>
{% endif %}
```

1.7.46.1.7. subscribable?

1.7.46.1.8. subscribe_url

1.7.46.1.9. application_plans

返回服务的 **published** 应用计划。

```
{% for service in model.services %}
  <h4>{{ service.name }} application plans:</h4>
  <dl>
    {% for application_plan in service.application_plans %}
      <dt>{{ application_plan.name }}</dt>
      <dd>{{ application_plan.system_name }}</dd>
    {% endfor %}
  </dl>
{% endfor %}
```

1.7.46.1.10. service_plans

返回服务的 *published* 服务计划。

```
<p>We offer following service plans:</p>
<dl>
  {% for service in model.services %}
    {% for service_plan in service.service_plans %}
      <dt>{{ service_plan.name }}</dt>
      <dd>{{ service_plan.system_name }}</dd>
```

```
{% endfor %}
{% endfor %}
</dl>
```

1.7.46.1.11. plans

返回服务的应用计划。

1.7.46.1.12. 功能

返回服务的可见功能。

```
{% if service.features.size > 0 %}
  <p>{{ service.name }} has the following features:</p>
  <ul>
    {% for feature in service.features %}
      <li>{{ feature.name }}</li>
    {% endfor %}
  </ul>
{% else %}
  <p>Unfortunately, {{ service.name }} currently has no features.</p>
{% endif %}
```

1.7.46.1.13. apps_identifier

根据设置的身份验证模式，返回用于 OAuth 身份验证的 'ID'、'API 密钥' 或 'Client ID'。

```
{{ service.application_key_name }}
```

1.7.46.1.14. backend_version

1.7.46.1.15. referencer_filters_required?

1.7.46.1.16. metrics

返回服务指标。

```
<p>On {{ service.name }} we measure the following metrics:</p>
<ul>
  {% for metric in service.metrics %}
    <li>{{ metric.name }}</li>
  {% endfor %}
</ul>
```

1.7.46.1.17. support_email

支持服务电子邮件。

1.7.47. ServiceContract drop (up)

1.7.47.1. Methods

1.7.47.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ service_contract.errors.name | inline_errors }}
```

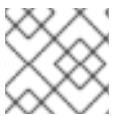
1.7.47.1.2. id

1.7.47.1.3. can_change_plan?

如果可能采用任何形式的更改，则返回 true。

1.7.47.1.4. trial?

如果合同仍在试用期间，返回真。



注意

如果您更改计划的试用期限，则不影响现有合同。

1.7.47.1.5. live?

1.7.47.1.6. state

有三种可能的状态：

- pending
- live
- suspended

1.7.47.1.7. remaining_trial_period_days

试用期间仍存在的天数。

1.7.47.1.8. plan

返回合同计划。

1.7.47.1.9. plan_change_permission_name

返回允许操作的名称。

1.7.47.1.10. plan_change_permission_warning

返回允许操作的警告消息。

1.7.47.1.11. 合同

1.7.47.1.12. name

1.7.47.1.13. system_name

1.7.47.1.14. change_plan_url

1.7.47.1.15. service

1.7.47.1.16. 应用程序

1.7.47.1.17. can

公开当前用户对该订阅的特定权限。

```
{% if subscription.can.change_plan? %}  
  ...  
{% endif %}
```

1.7.48. ServicePlan drop (up)

1.7.48.1. Methods

1.7.48.1.1. selected?

返回是否选择了计划。

```
{% if plan.selected? %}  
  <p>You will signup to {{ plan.name }}</p>  
{% endif %}
```

1.7.48.1.2. bought?

返回计划是否购买。

```
{% if plan.bought? %}  
  <p>You are on this plan already!</p>  
{% endif %}
```

1.7.48.1.3. 功能

返回计划可见的功能。

```
{% if plan == my_free_plan %}  
  <p>These plans are the same.</p>  
{% else %}  
  <p>These plans are not the same.</p>  
{% endif %}
```

1.7.48.1.4. setup_fee

返回计划的设置费用。

1.7.48.1.5. name

返回计划的名称。

```
<h2>We offer you a new {{ plan.name }} plan!</h2>
```

1.7.48.1.6. system_name

返回计划的系统名称。

```
{% for plan in available_plans %}
  {% if plan.system_name == 'my_free_plan' %}

    <p>You will buy our only free plan!</p>
  {% endif %}
{% endfor %}
```

1.7.48.1.7. id

返回计划 ID。

1.7.48.1.8. free?

如果计划不是 **paid**，则为免费计划。请参阅 *paid?* method.

```
{% if plan.free? %}
  <p>This plan is free of charge.</p>
{% else %}

  <p>Plan costs</p>
  Setup fee {{ plan.setup_fee }}
  Flat cost {{ plan.flat_cost }}

{% endif %}
```

1.7.48.1.9. paid?

当计划具有非零的固定或设置费用或存在一些定价规则时，计划为 **paid**。

```
{% if plan.paid? %}
  <p>this plan is a paid one.</p>
{% else %}
  <p>this plan is a free one.</p>
{% endif %}
```

1.7.48.1.10. approval_required?

返回计划是否需要审批？

```
{% if plan.approval_required? %}
  <p>This plan requires approval.</p>
{% endif %}
```

1.7.48.1.11. flat_cost

返回计划的月度固定费用。

1.7.48.1.12. service

示例： 使用服务计划丢弃。

```
<p class="notice">The examples for plan drop apply here</p>
Service of this plan {{ plan.service.name }}
```

1.7.49. TimeZone drop (up)

1.7.49.1. Methods

1.7.49.1.1. full_name

1.7.49.1.2. to_str

1.7.50. Today drop (up)

1.7.50.1. Methods

1.7.50.1.1. month

返回当前月份(1-12)。

1.7.50.1.2. day

返回当月的当前日期(1-31)。

1.7.50.1.3. year

返回当前年份。**示例：** 创建动态版权。

```
&copy;{{ today.year }}
```

1.7.50.1.4. beginning_of_month

当前月开始的返回日期。

```
This month began on {{ today.beginning_of_month | date: '%A' }}
```

1.7.51. Topic drop (up)

1.7.51.1. Methods

1.7.51.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ topic.errors.name | inline_errors }}
```

1.7.51.1.2. title

1.7.51.1.3. kind

1.7.51.1.4. url

1.7.51.1.5. description

1.7.52. Topic drop (up)

1.7.52.1. Methods

1.7.52.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ topic.errors.name | inline_errors }}
```

1.7.52.1.2. title

主题的名称。首次发布到线程时提交。

1.7.52.1.3. url

1.7.53. URL drop (up)

1.7.53.1. Methods

1.7.53.1.1. to_s

1.7.53.1.2. to_str

1.7.53.1.3. title

1.7.53.1.4. current_or_subpath?

如果当前页面的路径与 URL 的路径相同，则为 true，或者它是它的 *直接子路径*，即 ID 扩展。例如，对于 `{{ urls.outbox }}`，它们将返回 true：

- /admin/sent/messages/received
- /admin/sent/messages/received/42

但不是这些：

- /admin/sent/messages/new
- /admin/sent/messages/longer/subpath

另请参阅 '#active?', '#current?'。

1.7.53.1.5. current?

如果 URL 路径与当前路径相同，则为 true。不考虑参数和其他组件。另请参阅 [active?'。

```
{% assign url = urls.messages_inbox %}
<!-- => http://awesome.3scale.net/admin/messages/sent -->

<!-- Current page: http://awesome.3scale.net/admin/messages/sent?unread=1 -->
{{ url.current? }} => true

<!-- Current page: http://awesome.3scale.net/admin/messages -->
{{ url.current? }} => false
```

1.7.53.1.6. active?

如果当前页面位于与这个 URL 相同的菜单结构中则为 true。另请参阅 #current?

```
{% assign url = urls.messages_inbox %}
<!-- => http://awesome.3scale.net/admin/messages/sent -->

<!-- Current page: http://awesome.3scale.net/admin/messages -->
{{ url.active? }} => true

<!-- Current page: http://awesome.3scale.net/admin/messages/trash -->
{{ url.active? }} => true

<!-- Current page: http://awesome.3scale.net/admin/stats -->
{{ url.active? }} => false
```

1.7.54. Urls drop (up)

1.7.54.1. Methods

1.7.54.1.1. provider

1.7.54.1.2. cas_login

```
<a href="{{ urls.signup }}">signup here</a>
<a href="{{ urls.service_subscription }}">subscribe to a service here</a>
```

1.7.54.1.3. new_application

1.7.54.1.4. signup

注册页面的 URL。所有用户均可访问。

-

```
<a href="{{ urls.signup }}?{{ service_plan | param_filter }}&{{ app_plan | param_filter }}" >Signup Now!
</a>
```

1.7.54.1.5. search

所有搜索请求应发送到的 URL。

```
<form action="{{ urls.search }}" method="get">

</form>
```

1.7.54.1.6. login

1.7.54.1.7. logout

1.7.54.1.8. forgot_password

1.7.54.1.9. service_subscription

服务订阅页面的 URL。仅针对已登录用户。

```
<a href="{{ urls.service_subscription }}?{{ service_plan | param_filter }}" >
  Subscribe to service {{ service.name }}
</a>
```

1.7.54.1.10. compose_message

允许开发人员通过内部消息传递系统联系提供程序的页面 URL。

1.7.54.1.11. messages_outbox

开发人员发送的消息列表的 URL。

1.7.54.1.12. messages_trash

1.7.54.1.13. empty_messages_trash

1.7.54.1.14. credit_card_terms

1.7.54.1.15. credit_card_privacy

1.7.54.1.16. credit_card_refunds

1.7.54.1.17. users

1.7.54.1.18. personal_details

如果用户帐户管理链接禁用，则 URL 或 URL 不会在帐户管理页面的帐户管理部分中显示。

如果用户帐户管理被禁用，则 URL 或 Nil。在 3scale 管理门户的 **Audience > Accounts > Settings > Usage Rules** 下选择 **Usage Rules**。

1.7.54.1.19. access_details

包含 API 密钥和其他身份验证信息的页面。根据身份验证策略的不同而有所不同。

1.7.54.1.20. payment_details

1.7.54.1.21. new_invitation

邀请新用户的页面。

1.7.54.1.22. 邀请

所有发送的邀请列表。

1.7.54.1.23. dashboard

1.7.54.1.24. 应用程序

1.7.54.1.25. api_access_details

1.7.54.1.26. services

1.7.54.1.27. messages_inbox

收到的邮件列表的 URL。

1.7.54.1.28. stats

1.7.54.1.29. account_overview

1.7.54.1.30. account_plans

1.7.54.1.31. invoices

1.7.55. UsageLimit drop (up)

示例：在 liquid 中使用用量限制丢弃。

You cannot do more than `{{ limit.value }}` `{{ limit.metric.unit }}`s per `{{ limit.period }}`

1.7.55.1. Methods

1.7.55.1.1. 周期

返回用量限制的期间。

1.7.55.1.2. metric

通常会 **hits**，但可以是任何自定义方法。

1.7.55.1.3. value

返回用量限制的值。

1.7.56. User drop (up)

```
<h2>User {{ user.display_name }}</h2>
Account {{ user.account.name }}
Username {{ user.username }}
Email {{ user.email }}
Website {{ user.website }}
```

1.7.56.1. Methods

1.7.56.1.1. errors

如果在提交失败后呈现此模型，这将返回发生的错误。

```
{{ user.errors.name | inline_errors }}
```

1.7.56.1.2. admin?

返回用户是否为 admin。

```
{% if user.admin? %}
  <p>You are an admin of your account.</p>
{% endif %}
```

1.7.56.1.3. username

返回用户的用户名（html 转意）。

1.7.56.1.4. account

返回用户的帐户。

1.7.56.1.5. name

返回用户的名和姓。

1.7.56.1.6. email

返回用户的电子邮件。

1.7.56.1.7. password_required?

对于使用内置的 Developer Portal 认证机制的用户返回 **true**，对于经过 Janrain, CAS 或其他单点登陆方法的用户返回 **false**。

```
{% if user.password_required? %}

{% endif %}
```

1.7.56.1.8. 部分

返回用户有权访问的部分列表。

```
{% if user.sections.size > 0 %}
<p>You can access following sections of our portal:</p>
<ul>
  {% for section in user.sections %}
    <li>{{ section }}</li>
  {% endfor %}
</ul>
{% endif %}
```

1.7.56.1.9. role

返回用户的角色。

1.7.56.1.10. roles_collection

重新调整用户的可用角色列表。

```
{% for role in user.roles_collection %}
<li>
  <label for="user_role_{{ role.key }}">

    {{ role.text&nbsp; }}
  </label>
</li>
{% endfor %}
```

1.7.56.1.11. url

返回用户的资源 URL。

```
{{ 'Delete' | delete_button: user.url }}
```

1.7.56.1.12. edit_url

返回 URL 以编辑用户。

```
{{ 'Edit' | link_to: user.edit_url, title: 'Edit', class: 'action edit' }}
```

1.7.56.1.13. can

授予对权限方法的访问权限。


```
{% if user.can.be_managed? %}
  <!-- do something -->
{% endif %}
```

1.7.56.114. extra_fields

返回带有此用户值的非隐藏的额外字段。**示例**：打印所有额外字段。

```
{% for field in user.extra_fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.56.115. fields

返回所有带有此用户值的字段。**示例**：打印所有字段。

```
{% for field in user.fields %}
  {{ field.label }}: {{ field.value }}
{% endfor %}
```

1.7.56.116. builtin_fields

返回所有带有此用户值的内置字段。

1.8. TAGS (UP)

1.8.1. Tag 'braintree_customer_form' (up)

呈现输入 Braintree Blue 支付网关所需数据的表单。

1.8.2. Tag 'csrf' (up)

呈现跨站点请求伪造元标签。

示例：在 liquid 中使用 csrf 标签。

```
<head>
  {% csrf %}
</head>
```

1.8.3. Tag 'content' (up)

呈现页面的正文.仅在布局内使用。

1.8.4. Tag 'content_for' (up)

1.8.5. Tag 'debug' (up)

将模板中所有可用的变量打印到 HTML 注释中。我们建议 **从公共模板中删除此标签**。

```
`{% debug:help %}`
```

1.8.6. Tag 'email' (up)

电子邮件 标签允许您自定义传出电子邮件的标头，且仅在电子邮件模板中可用。

有几个方便的子标签，如 **cc** 或 **subject**（请参见下表），但您也可以使用 **header** 子标签为邮件设置任意 SMTP 标头。

子标签	描述	示例
subject	动态主题	<code>{% subject = 'Greetings from Example company!' %}</code>
cc	抄送	<code>{% cc = 'boss@example.com' %}</code>
bcc	秘密抄送	<code>{% bcc = 'all@example.com' %}</code>
from	实际发送者	<code>{% from = 'system@example.com' %}</code>
reply-to		<code>{% reply-to = 'support@example.com' %}</code>
header	自定义 SMTP 标头	<code>{% header 'X-SMTP-Group' = 'Important' %}</code>
do_not_send	丢弃电子邮件	<code>{% do_not_send %}</code>

示例： 有条件的 blind carbon 复制。

```
{% email %}
  {% if plan.system_name == 'enterprise' %}
    {% bcc 'marketing@world-domination.org' %}
  {% endif %}
{% endemail %}
```

示例： 完全禁用电子邮件。

```
{% email %}
  {% do_not_send %}
{% endemail %}
```

示例： Signup email filter

```
{% email %}
  {% if plan.system == 'enterprise' %}
    {% subject = 'Greetings from Example company!' %}
    {% reply-to = 'support@example.com' %}
  {% else %}
```

```
{% do_not_send %}
{% endif %}
{% endemail %}
```

1.8.7. Tag 'flash' (up)

呈现系统的信息或错误消息。

DEPRECATED: 此标签已弃用，改为使用 FlashDrop。

示例： 在 liquid 中使用 flash 标签。

```
{% flash %}
```

1.8.8. Tag 'footer' (up)

呈现页脚 html 代码片段。

DEPRECATED : 此标签已弃用。而是使用开发人员门户部分。

1.8.9. Tag 'form' (up)

根据表单的名称，呈现带有指定操作和类属性的格式标签。支持的表单有：

格式	允许的字段名称	垃圾邮件保护	备注
application.create	<ul style="list-style-type: none"> ● application[name] ● application[description] ● application[<any-extra-field>] 	否	
application.update	<ul style="list-style-type: none"> ● application[name] ● application[description] ● application[<any-extra-field>] 	否	

signup	<ul style="list-style-type: none"> ● account[org_name] ● account[org_legaladdress] ● account[org_legaladdress_content] ● account[city] ● account[state] ● account[zip] ● account[telephone_number] ● account[country_id] ● account[<any-extra-field>] ● account[user][username] ● account[user][email] ● account[user][first_name] ● account[user][last_name] ● account[user][password] ● account[user][password_confirmation] ● account[user][title] ● account[user][<any-extra-field>] 	是	<p>通过添加一个或多个带有名称 plan_ids[] 的隐藏字段，直接登录到您选择的计划。如果在当前 URL 中找到此类名称的参数，则会自动填充输入字段。</p>
--------	---	---	---

示例： 创建应用程序的表单。

```
{% form 'application.create', application %}

  {{ application.errors.name | inline_errors }}

{% endform %}
```

1.8.10. Tag 'latest_forum_posts' (up)

包含最新论坛文章的 HTML 表。

DEPRECATED: 使用 **forum** drop 替代。

示例： 使用 latest_forum_posts tag liquid.

```
{% latest_forum_posts %}
```

1.8.11. Tag 'latest_messages' (up)

呈现包含用户最新消息的 HTML 代码片段。

示例： 使用 latest_messages 标签 liquid.

```
{% latest_messages %}
```

1.8.12. Tag 'logo' (up)

呈现徽标。

DEPRECATED: 此标签已弃用，改为使用 `{{ provider.logo_url }}`。

示例： 在 liquid 中使用菜单标签。

```
{% logo %}
```

1.8.13. Tag 'menu' (up)

DEPRECATED: 此标签已弃用，改为使用 `'{% include "menu" %}'`。

1.8.14. Tag 'oldfooter' (up)

呈现页脚 html 代码片段。

DEPRECATED : 此标签已弃用。而是使用开发人员门户部分。

1.8.15. Tag 'plan_widget' (up)

包含用于检查或更改应用计划的小部件。

```
{% if application.can_change_plan? %}
<a href="#choose-plan-{{ application.id }}"
  id="choose-plan-{{application.id}}">
  Review/Change
</a>
{% plan_widget application %}
{% endif %}
```

1.8.16. Tag 'portlet' (up)

此标签包含按系统名称的 portlet。

1.8.17. Tag 'submenu' (up)

为已登录用户呈现子菜单 html 代码片段。

DEPRECATED: 此标签已弃用，改为使用 'submenu' 部分。

示例： 在 liquid 中使用 submenu 标签。

```
{% submenu %}
```

1.8.18. Tag '3scale_essentials' (up)

1.8.19. Tag 'user_widget' (up)

呈现用户小部件 html 代码段。

DEPRECATED : 此标签已弃用。而是使用开发人员门户部分。

示例： 在 liquid 中使用 user_widget 标签。

```
{% user_widget %}
<p class="notice">If you are logged in you see profile related links above.</p>
<p class="notice">If you are not login you are invited to login or signup.</p>
```

1.9. FILTERS (UP)

1.9.1. FormHelpers filters (up)

1.9.1.1. error_class filter

如果参数不为空，则输出错误类。**示例：** 使用 error_class 显示输出错误类。

1.9.1.2. inline_errors 过滤器

输出段落中内联的错误字段。**示例：** 使用 inline_errors 显示内联错误。

```
{{ form.errors.description | inline_errors }}
```

1.9.2. ParamFilter filters (up)

1.9.2.1. to_param filter

将提供的 drop（如果可能）转换为 URL 参数。**示例：** 在 liquid 中使用 to_param 过滤器。

```
<h2>Signup to a service</h2>
<a href="{{ urls.signup }}?{{ service | to_param }}">Signup to {{ service.name }}</a>
```

1.9.3. 常用过滤器 (up)

1.9.3.1. group_by 过滤器

由某些键组成的组集合。示例：按服务分组应用程序。

```
{% assign grouped = applications | group_by: 'service' %}
{% for group in grouped %}
  Service: {{ group[0] }}
  {% for app in group[1] %}
    Application: {{ app.name }}
  {% endfor %}
{% endfor %}
```

1.9.3.2. 任何过滤器

如果集合中的任何字符串等于参数，则为 true。示例：当前帐户是否有待处理的应用程序？

```
{% assign has_pending_apps = current_account.applications | map: 'state' | any: 'live' %}
```

1.9.3.3. stylesheet_link_tag filter

Stylesheet 链接

1.9.3.4. javascript_include_tag 过滤器

JavaScript 包含标签

1.9.3.5. IMAGE_TAG 过滤器

使用参数输出标签作为其 **src** 属性。

```
{{ 'http://example.com/cool.gif' | image_tag }}
## =>
```

1.9.3.6. mail_to 过滤器

将电子邮件地址转换为 'mailto' 链接。

```
{{ 'me@there.is' | mail_to }}
## => <a href="mailto:me@there.is">me@there.is</a>
```

1.9.3.7. html_safe filter

将内容标记为 HTML 安全，以免内容被转义。

1.9.3.8. pluralize filter

将单词转换为复数形式

1.9.3.9. delete_button filter

生成按钮以删除 URL 上存在的资源。第一个参数是 URL，第二个是标题。您还可以添加更多 HTML 标签属性作为第三个参数。

```
{{ 'Delete Message' | delete_button: message.url, class: 'my-button' }}
```

1.9.3.10. delete_button_ajax filter

生成按钮，以使用 AJAX 删除 URL 上存在的资源。第一个参数是 URL，第二个是标题。

```
{{ 'Delete Message' | delete_button_ajax: message.url }}
```

1.9.3.11. update_button filter

生成一个按钮到"更新" (HTTP PUT 请求)，某个资源存在于 URL 上。第一个参数是 URL，第二个是标题。您还可以添加更多 HTML 标签属性作为第三个参数。

```
{{ 'Resend' | update_button: message.url, class: 'my-button' }}
```

1.9.3.12. create_button filter

生成按钮以创建 URL 上存在的资源。第一个参数是 URL，第二个是标题。

```
{{ 'Create Message' | create_button: message.url }}
```

1.9.3.13. create_button_ajax filter

1.9.3.14. regenerate_oauth_secret_button filter

1.9.3.15. link_to filter

从给定文本创建链接。

```
{{ "See your App keys" | link_to: '/my-app-keys' }}
```