



Red Hat 3scale API Management 2.13

Liquid Reference

Find additional information related to your 3scale installation.

Red Hat 3scale API Management 2.13 Liquid Reference

Find additional information related to your 3scale installation.

Legal Notice

Copyright © 2022 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.

Abstract

This guide provides the information about components that interact with your 3scale installation.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	15
CHAPTER 1. LIQUID REFERENCE	16
1.1. WHAT ARE LIQUIDS?	16
1.1.1. Pros and cons	16
1.2. HOW TO USE LIQUIDS	16
1.2.1. Liquid drops, tags, and filters	16
1.2.2. The context	17
1.2.3. Hierarchy	17
1.3. LIQUID TEMPLATES	17
1.4. DROPS	18
1.5. TAGS	20
1.6. FILTERS	20
1.7. DROPS (UP)	21
1.7.1. Account drop (up)	21
1.7.1.1. Methods	21
1.7.1.1.1. errors	21
1.7.1.1.2. id	21
1.7.1.1.3. name	21
1.7.1.1.4. vat_zero_text	22
1.7.1.1.5. vat_rate	22
1.7.1.1.6. unread_messages	22
1.7.1.1.7. latest_messages	22
1.7.1.1.8. bought_account_plan	22
1.7.1.1.9. bought_account_contract	22
1.7.1.1.10. credit_card_display_number	22
1.7.1.1.11. credit_card_expiration_date	22
1.7.1.1.12. credit_card_required?	22
1.7.1.1.13. credit_card_stored?	22
1.7.1.1.14. credit_card_missing?	22
1.7.1.1.15. timezone	22
1.7.1.1.16. paid?	22
1.7.1.1.17. on_trial?	22
1.7.1.1.18. telephone_number	22
1.7.1.1.19. approval_required?	23
1.7.1.1.20. created_at	23
1.7.1.1.21. full_address	23
1.7.1.1.22. applications	23
1.7.1.1.23. subscribed_services	23
1.7.1.1.24. admin	23
1.7.1.1.25. extra_fields_plain_text	23
1.7.1.1.26. fields_plain_text	23
1.7.1.1.27. extra_fields	23
1.7.1.1.28. fields	23
1.7.1.1.29. builtin_fields	24
1.7.1.1.30. multiple_applications_allowed?	24
1.7.1.1.31. billing_address	24
1.7.1.1.32. has_billing_address?	24
1.7.1.1.33. can	24
1.7.1.1.34. edit_url	24
1.7.1.1.35. edit_ogone_billing_address_url	24

1.7.1.1.36. edit_payment_express_billing_address_url	24
1.7.1.1.37. edit_braintree_blue_credit_card_details_url	24
1.7.1.1.38. domain	24
1.7.1.1.39. upgraded?	24
1.7.1.1.40. requires_credit_card?	24
1.7.1.1.41. support_email	24
1.7.1.1.42. finance_support_email	24
1.7.2. AccountPlan drop (up)	24
1.7.2.1. Methods	25
1.7.2.1.1. selected?	25
1.7.2.1.2. bought?	25
1.7.2.1.3. features	25
1.7.2.1.4. setup_fee	25
1.7.2.1.5. name	25
1.7.2.1.6. system_name	25
1.7.2.1.7. id	25
1.7.2.1.8. free?	25
1.7.2.1.9. paid?	26
1.7.2.1.10. approval_required?	26
1.7.2.1.11. flat_cost	26
1.7.3. Alert drop (up)	26
1.7.3.1. Methods	26
1.7.3.1.1. level	26
1.7.3.1.2. message	26
1.7.3.1.3. utilization	27
1.7.4. Application drop (up)	27
1.7.4.1. Methods	27
1.7.4.1.1. errors	27
1.7.4.1.2. id	27
1.7.4.1.3. can_change_plan?	27
1.7.4.1.4. trial?	27
1.7.4.1.5. live?	27
1.7.4.1.6. state	27
1.7.4.1.7. remaining_trial_period_days	27
1.7.4.1.8. plan	27
1.7.4.1.9. plan_change_permission_name	28
1.7.4.1.10. plan_change_permission_warning	28
1.7.4.1.11. contract	28
1.7.4.1.12. admin_url	28
1.7.4.1.13. name	28
1.7.4.1.14. can	28
1.7.4.1.15. oauth	28
1.7.4.1.16. pending?	28
1.7.4.1.17. buyer_alerts_enabled?	28
1.7.4.1.18. description	28
1.7.4.1.19. redirect_url	28
1.7.4.1.20. filters_limit	28
1.7.4.1.21. keys_limit	28
1.7.4.1.22. referrer_filters	28
1.7.4.1.23. rejection_reason	28
1.7.4.1.24. user_key	29
1.7.4.1.25. application_id	29
1.7.4.1.26. key	29

1.7.4.1.27. url	29
1.7.4.1.28. edit_url	29
1.7.4.1.29. update_user_key_url	29
1.7.4.1.30. log_requests_url	29
1.7.4.1.31. alerts_url	29
1.7.4.1.32. application_keys_url	29
1.7.4.1.33. service	29
1.7.4.1.34. keys	29
1.7.4.1.35. user_key_mode?	30
1.7.4.1.36. app_id_mode?	30
1.7.4.1.37. change_plan_url	30
1.7.4.1.38. log_requests?	30
1.7.4.1.39. application_keys	30
1.7.4.1.40. extra_fields	30
1.7.4.1.41. fields	30
1.7.4.1.42. builtin_fields	30
1.7.4.1.43. cinstance	30
1.7.5. ApplicationKey drop (up)	30
1.7.5.1. Methods	30
1.7.5.1.1. id	30
1.7.5.1.2. value	30
1.7.5.1.3. url	30
1.7.5.1.4. application	30
1.7.6. ApplicationPlan drop (up)	30
1.7.6.1. Methods	30
1.7.6.1.1. selected?	31
1.7.6.1.2. bought?	31
1.7.6.1.3. features	31
1.7.6.1.4. setup_fee	31
1.7.6.1.5. name	31
1.7.6.1.6. system_name	31
1.7.6.1.7. id	31
1.7.6.1.8. free?	32
1.7.6.1.9. paid?	32
1.7.6.1.10. approval_required?	32
1.7.6.1.11. flat_cost	32
1.7.6.1.12. metrics	32
1.7.6.1.13. usage_limits	32
1.7.6.1.14. service	32
1.7.7. Base drop (up)	32
1.7.7.1. Methods	32
1.7.7.1.1. login_url	33
1.7.7.1.2. user_identified?	33
1.7.8. Base drop (up)	33
1.7.8.1. Methods	33
1.7.8.1.1. errors	33
1.7.8.1.2. title	33
1.7.8.1.3. kind	33
1.7.8.1.4. url	33
1.7.8.1.5. description	33
1.7.9. BillingAddressField drop (up)	33
1.7.9.1. Methods	33
1.7.9.1.1. input_name	33

1.7.9.1.2. label	33
1.7.9.1.3. choices	33
1.7.9.1.4. errors	33
1.7.9.1.5. html_id	33
1.7.9.1.6. hidden?	33
1.7.9.1.7. visible?	33
1.7.9.1.8. read_only?	34
1.7.9.1.9. name	34
1.7.9.1.10. value	34
1.7.9.1.11. required	34
1.7.10. Can drop (up)	34
1.7.10.1. Methods	34
1.7.10.1.1. be_updated?	34
1.7.10.1.2. add_referrer_filters?	34
1.7.10.1.3. add_application_keys?	34
1.7.10.1.4. regenerate_user_key?	34
1.7.10.1.5. regenerate_oauth_secret?	34
1.7.10.1.6. manage_keys?	34
1.7.10.1.7. delete_key?	34
1.7.11. Can drop (up)	34
1.7.11.1. Methods	34
1.7.11.1.1. change_plan?	34
1.7.12. Cas drop (up)	34
1.7.12.1. Methods	34
1.7.12.1.1. login_url	34
1.7.12.1.2. user_identified?	34
1.7.13. Contract drop (up)	34
1.7.13.1. Methods	34
1.7.13.1.1. errors	35
1.7.13.1.2. id	35
1.7.13.1.3. can_change_plan?	35
1.7.13.1.4. trial?	35
1.7.13.1.5. live?	35
1.7.13.1.6. state	35
1.7.13.1.7. remaining_trial_period_days	35
1.7.13.1.8. plan	35
1.7.13.1.9. plan_change_permission_name	35
1.7.13.1.10. plan_change_permission_warning	35
1.7.13.1.11. contract	35
1.7.14. Country drop (up)	35
1.7.14.1. Methods	35
1.7.14.1.1. errors	36
1.7.14.1.2. to_str	36
1.7.14.1.3. code	36
1.7.14.1.4. label	36
1.7.15. CountryField drop (up)	36
1.7.15.1. Methods	36
1.7.15.1.1. value	36
1.7.15.1.2. name	36
1.7.15.1.3. required	36
1.7.15.1.4. hidden?	36
1.7.15.1.5. hidden	36
1.7.15.1.6. visible?	36

1.7.15.1.7. visible	36
1.7.15.1.8. read_only	36
1.7.15.1.9. errors	36
1.7.15.1.10. input_name	36
1.7.15.1.11. html_id	36
1.7.15.1.12. label	37
1.7.15.1.13. to_str	37
1.7.15.1.14. choices	37
1.7.16. CurrentUser drop (up)	37
1.7.16.1. Methods	37
1.7.16.1.1. errors	37
1.7.16.1.2. admin?	37
1.7.16.1.3. username	37
1.7.16.1.4. account	37
1.7.16.1.5. name	37
1.7.16.1.6. email	37
1.7.16.1.7. password_required?	37
1.7.16.1.8. sections	38
1.7.16.1.9. role	38
1.7.16.1.10. roles_collection	38
1.7.16.1.11. url	38
1.7.16.1.12. edit_url	38
1.7.16.1.13. can	38
1.7.16.1.14. extra_fields	39
1.7.16.1.15. fields	39
1.7.16.1.16. builtin_fields	39
1.7.17. Error drop (up)	39
1.7.17.1. Methods	39
1.7.17.1.1. attribute	39
1.7.17.1.2. message	39
1.7.17.1.3. value	40
1.7.17.1.4. to_str	40
1.7.18. Errors drop (up)	40
1.7.18.1. Methods	40
1.7.18.1.1. empty?	40
1.7.18.1.2. present?	40
1.7.19. Feature drop (up)	40
1.7.19.1. Methods	40
1.7.19.1.1. errors	40
1.7.19.1.2. name	41
1.7.19.1.3. description	41
1.7.19.1.4. has_description?	41
1.7.20. Field drop (up)	41
1.7.20.1. Methods	41
1.7.20.1.1. value	41
1.7.20.1.2. name	41
1.7.20.1.3. required	41
1.7.20.1.4. hidden?	41
1.7.20.1.5. hidden	41
1.7.20.1.6. visible?	41
1.7.20.1.7. visible	41
1.7.20.1.8. read_only	41
1.7.20.1.9. errors	42

1.7.20.1.10. input_name	42
1.7.20.1.11. html_id	42
1.7.20.1.12. label	42
1.7.20.1.13. to_str	42
1.7.20.1.14. choices	42
1.7.21. Flash drop (up)	42
1.7.21.1. Methods	42
1.7.21.1.1. messages	43
1.7.22. Forum drop (up)	43
1.7.22.1. Methods	43
1.7.22.1.1. enabled?	43
1.7.22.1.2. latest_posts	43
1.7.23. I18n drop (up)	43
1.7.23.1. Methods	43
1.7.23.1.1. short_date	43
1.7.23.1.2. long_date	43
1.7.23.1.3. default_date	43
1.7.24. Invitation drop (up)	44
1.7.24.1. Methods	44
1.7.24.1.1. errors	44
1.7.24.1.2. email	44
1.7.24.1.3. accepted?	44
1.7.24.1.4. accepted_at	44
1.7.24.1.5. sent_at	44
1.7.24.1.6. resend_url	44
1.7.24.1.7. url	45
1.7.25. Invoice drop (up)	45
1.7.25.1. Methods	45
1.7.25.1.1. errors	45
1.7.25.1.2. friendly_id	45
1.7.25.1.3. name	45
1.7.25.1.4. state	45
1.7.25.1.5. cost	45
1.7.25.1.6. currency	45
1.7.25.1.7. cost_without_vat	45
1.7.25.1.8. vat_amount	45
1.7.25.1.9. exists_pdf?	45
1.7.25.1.10. period_begin	46
1.7.25.1.11. period_end	46
1.7.25.1.12. issued_on	46
1.7.25.1.13. due_on	46
1.7.25.1.14. paid_on	46
1.7.25.1.15. vat_code	46
1.7.25.1.16. fiscal_code	46
1.7.25.1.17. account	46
1.7.25.1.18. buyer_account	46
1.7.25.1.19. line_items	46
1.7.25.1.20. payment_transactions	46
1.7.25.1.21. url	47
1.7.25.1.22. pdf_url	47
1.7.26. Janrain drop (up)	47
1.7.26.1. Methods	47
1.7.26.1.1. login_url	47

1.7.26.1.2. user_identified?	47
1.7.26.1.3. session_url	47
1.7.26.1.4. relying_party	47
1.7.27. Lineltem drop (up)	47
1.7.27.1. Methods	47
1.7.27.1.1. errors	47
1.7.27.1.2. name	47
1.7.27.1.3. description	48
1.7.27.1.4. quantity	48
1.7.27.1.5. cost	48
1.7.28. Message drop (up)	48
1.7.28.1. Methods	48
1.7.28.1.1. errors	48
1.7.28.1.2. id	48
1.7.28.1.3. subject	48
1.7.28.1.4. body	48
1.7.28.1.5. created_at	48
1.7.28.1.6. url	48
1.7.28.1.7. state	48
1.7.28.1.8. sender	48
1.7.28.1.9. to	49
1.7.28.1.10. recipients	49
1.7.29. Message drop (up)	49
1.7.29.1. Methods	49
1.7.29.1.1. type	49
1.7.29.1.2. text	49
1.7.30. Metric drop (up)	49
1.7.30.1. Methods	49
1.7.30.1.1. errors	49
1.7.30.1.2. unit	49
1.7.30.1.3. description	49
1.7.30.1.4. name	49
1.7.30.1.5. system_name	50
1.7.30.1.6. usage_limits	50
1.7.30.1.7. pricing_rules	50
1.7.30.1.8. has_parent	50
1.7.31. Page drop (up)	50
1.7.31.1. Methods	50
1.7.31.1.1. errors	50
1.7.31.1.2. title	50
1.7.31.1.3. system_name	51
1.7.32. Page drop (up)	51
1.7.32.1. Methods	51
1.7.32.1.1. errors	51
1.7.32.1.2. title	51
1.7.32.1.3. kind	51
1.7.32.1.4. url	51
1.7.32.1.5. description	51
1.7.33. Pagination drop (up)	51
1.7.33.1. Methods	51
1.7.33.1.1. page_size	51
1.7.33.1.2. current_page	52
1.7.33.1.3. current_offset	52

1.7.33.1.4. pages	52
1.7.33.1.5. items	52
1.7.33.1.6. previous	52
1.7.33.1.7. next	53
1.7.33.1.8. parts	53
1.7.34. Part drop (up)	53
1.7.34.1. Methods	53
1.7.34.1.1. url	53
1.7.34.1.2. rel	53
1.7.34.1.3. current?	53
1.7.34.1.4. is_link	53
1.7.34.1.5. title	53
1.7.34.1.6. to_s	53
1.7.35. PaymentGateway drop (up)	53
1.7.35.1. Methods	53
1.7.35.1.1. braintree_blue?	53
1.7.35.1.2. authorize_net?	53
1.7.35.1.3. type	53
1.7.36. PaymentTransaction drop (up)	53
1.7.36.1. Methods	53
1.7.36.1.1. errors	53
1.7.36.1.2. currency	54
1.7.36.1.3. amount	54
1.7.36.1.4. created_at	54
1.7.36.1.5. success?	54
1.7.36.1.6. message	54
1.7.36.1.7. reference	54
1.7.37. PlanFeature drop (up)	54
1.7.37.1. Methods	54
1.7.37.1.1. errors	54
1.7.37.1.2. name	54
1.7.37.1.3. description	55
1.7.37.1.4. has_description?	55
1.7.37.1.5. enabled?	55
1.7.38. Post drop (up)	55
1.7.38.1. Methods	55
1.7.38.1.1. errors	55
1.7.38.1.2. body	55
1.7.38.1.3. topic	55
1.7.38.1.4. created_at	55
1.7.38.1.5. url	55
1.7.39. Post drop (up)	55
1.7.39.1. Methods	55
1.7.39.1.1. errors	56
1.7.39.1.2. title	56
1.7.39.1.3. kind	56
1.7.39.1.4. url	56
1.7.39.1.5. description	56
1.7.40. PricingRule drop (up)	56
1.7.40.1. Methods	56
1.7.40.1.1. cost_per_unit	56
1.7.40.1.2. min	56
1.7.40.1.3. max	56

1.7.40.1.4. plan	56
1.7.41. Provider drop (up)	56
1.7.41.1. Methods	56
1.7.41.1.1. name	56
1.7.41.1.2. payment_gateway	57
1.7.41.1.3. domain	57
1.7.41.1.4. timezone	57
1.7.41.1.5. support_email	57
1.7.41.1.6. finance_support_email	57
1.7.41.1.7. telephone_number	57
1.7.41.1.8. multiple_applications_allowed?	57
1.7.41.1.9. logo_url	58
1.7.41.1.10. multiple_services_allowed?	58
1.7.41.1.11. finance_allowed?	58
1.7.41.1.12. multiple_users_allowed?	58
1.7.41.1.13. account_plans	58
1.7.41.1.14. services	58
1.7.41.1.15. signups_enabled?	59
1.7.41.1.16. account_management_enabled?	59
1.7.42. ReferrerFilter drop (up)	59
1.7.42.1. Methods	59
1.7.42.1.1. id	59
1.7.42.1.2. value	59
1.7.42.1.3. delete_url	59
1.7.42.1.4. application	59
1.7.43. Request drop (up)	59
1.7.43.1. Methods	59
1.7.43.1.1. request_uri	59
1.7.43.1.2. host_with_port	59
1.7.43.1.3. host	59
1.7.43.1.4. path	60
1.7.44. Role drop (up)	60
1.7.44.1. Methods	60
1.7.44.1.1. name	60
1.7.44.1.2. description	60
1.7.45. Search drop (up)	60
1.7.45.1. Methods	60
1.7.45.1.1. errors	60
1.7.45.1.2. query	60
1.7.45.1.3. total_found	61
1.7.45.1.4. results	61
1.7.46. Service drop (up)	61
1.7.46.1. Methods	61
1.7.46.1.1. errors	61
1.7.46.1.2. name	61
1.7.46.1.3. system_name	61
1.7.46.1.4. description	61
1.7.46.1.5. subscribed?	61
1.7.46.1.6. subscription	61
1.7.46.1.7. subscribable?	62
1.7.46.1.8. subscribe_url	62
1.7.46.1.9. application_plans	62
1.7.46.1.10. service_plans	62

1.7.46.1.11. plans	62
1.7.46.1.12. features	62
1.7.46.1.13. apps_identifier	63
1.7.46.1.14. backend_version	63
1.7.46.1.15. referrer_filters_required?	63
1.7.46.1.16. metrics	63
1.7.46.1.17. support_email	63
1.7.47. ServiceContract drop (up)	63
1.7.47.1. Methods	63
1.7.47.1.1. errors	63
1.7.47.1.2. id	63
1.7.47.1.3. can_change_plan?	63
1.7.47.1.4. trial?	63
1.7.47.1.5. live?	64
1.7.47.1.6. state	64
1.7.47.1.7. remaining_trial_period_days	64
1.7.47.1.8. plan	64
1.7.47.1.9. plan_change_permission_name	64
1.7.47.1.10. plan_change_permission_warning	64
1.7.47.1.11. contract	64
1.7.47.1.12. name	64
1.7.47.1.13. system_name	64
1.7.47.1.14. change_plan_url	64
1.7.47.1.15. service	64
1.7.47.1.16. applications	64
1.7.47.1.17. can	64
1.7.48. ServicePlan drop (up)	64
1.7.48.1. Methods	65
1.7.48.1.1. selected?	65
1.7.48.1.2. bought?	65
1.7.48.1.3. features	65
1.7.48.1.4. setup_fee	65
1.7.48.1.5. name	65
1.7.48.1.6. system_name	65
1.7.48.1.7. id	65
1.7.48.1.8. free?	66
1.7.48.1.9. paid?	66
1.7.48.1.10. approval_required?	66
1.7.48.1.11. flat_cost	66
1.7.48.1.12. service	66
1.7.49. TimeZone drop (up)	66
1.7.49.1. Methods	66
1.7.49.1.1. full_name	66
1.7.49.1.2. to_str	67
1.7.50. Today drop (up)	67
1.7.50.1. Methods	67
1.7.50.1.1. month	67
1.7.50.1.2. day	67
1.7.50.1.3. year	67
1.7.50.1.4. beginning_of_month	67
1.7.51. Topic drop (up)	67
1.7.51.1. Methods	67
1.7.51.1.1. errors	67

1.7.51.1.2. title	67
1.7.51.1.3. kind	67
1.7.51.1.4. url	67
1.7.51.1.5. description	67
1.7.52. Topic drop (up)	67
1.7.52.1. Methods	67
1.7.52.1.1. errors	68
1.7.52.1.2. title	68
1.7.52.1.3. url	68
1.7.53. Url drop (up)	68
1.7.53.1. Methods	68
1.7.53.1.1. to_s	68
1.7.53.1.2. to_str	68
1.7.53.1.3. title	68
1.7.53.1.4. current_or_subpath?	68
1.7.53.1.5. current?	68
1.7.53.1.6. active?	69
1.7.54. Urls drop (up)	69
1.7.54.1. Methods	69
1.7.54.1.1. provider	69
1.7.54.1.2. cas_login	69
1.7.54.1.3. new_application	69
1.7.54.1.4. signup	69
1.7.54.1.5. search	69
1.7.54.1.6. login	69
1.7.54.1.7. logout	69
1.7.54.1.8. forgot_password	69
1.7.54.1.9. service_subscription	70
1.7.54.1.10. compose_message	70
1.7.54.1.11. messages_outbox	70
1.7.54.1.12. messages_trash	70
1.7.54.1.13. empty_messages_trash	70
1.7.54.1.14. credit_card_terms	70
1.7.54.1.15. credit_card_privacy	70
1.7.54.1.16. credit_card_refunds	70
1.7.54.1.17. users	70
1.7.54.1.18. personal_details	70
1.7.54.1.19. access_details	70
1.7.54.1.20. payment_details	70
1.7.54.1.21. new_invitation	70
1.7.54.1.22. invitations	70
1.7.54.1.23. dashboard	70
1.7.54.1.24. applications	70
1.7.54.1.25. api_access_details	71
1.7.54.1.26. services	71
1.7.54.1.27. messages_inbox	71
1.7.54.1.28. stats	71
1.7.54.1.29. account_overview	71
1.7.54.1.30. account_plans	71
1.7.54.1.31. invoices	71
1.7.55. UsageLimit drop (up)	71
1.7.55.1. Methods	71
1.7.55.1.1. period	71

1.7.55.1.2. metric	71
1.7.55.1.3. value	71
1.7.56. User drop (up)	71
1.7.56.1. Methods	71
1.7.56.1.1. errors	71
1.7.56.1.2. admin?	72
1.7.56.1.3. username	72
1.7.56.1.4. account	72
1.7.56.1.5. name	72
1.7.56.1.6. email	72
1.7.56.1.7. password_required?	72
1.7.56.1.8. sections	72
1.7.56.1.9. role	72
1.7.56.1.10. roles_collection	73
1.7.56.1.11. url	73
1.7.56.1.12. edit_url	73
1.7.56.1.13. can	73
1.7.56.1.14. extra_fields	73
1.7.56.1.15. fields	73
1.7.56.1.16. builtin_fields	73
1.8. TAGS (UP)	74
1.8.1. Tag 'braintree_customer_form' (up)	74
1.8.2. Tag 'csrf' (up)	74
1.8.3. Tag 'content' (up)	74
1.8.4. Tag 'content_for' (up)	74
1.8.5. Tag 'debug' (up)	74
1.8.6. Tag 'email' (up)	74
1.8.7. Tag 'flash' (up)	75
1.8.8. Tag 'footer' (up)	75
1.8.9. Tag 'form' (up)	76
1.8.10. Tag 'latest_forum_posts' (up)	78
1.8.11. Tag 'latest_messages' (up)	78
1.8.12. Tag 'logo' (up)	78
1.8.13. Tag 'menu' (up)	78
1.8.14. Tag 'oldfooter' (up)	78
1.8.15. Tag 'plan_widget' (up)	78
1.8.16. Tag 'portlet' (up)	79
1.8.17. Tag 'submenu' (up)	79
1.8.18. Tag '3scale_essentials' (up)	79
1.8.19. Tag 'user_widget' (up)	79
1.9. FILTERS (UP)	79
1.9.1. FormHelpers filters (up)	79
1.9.1.1. error_class filter	79
1.9.1.2. inline_errors filter	79
1.9.2. ParamFilter filters (up)	79
1.9.2.1. to_param filter	80
1.9.3. Common filters (up)	80
1.9.3.1. group_by filter	80
1.9.3.2. any filter	80
1.9.3.3. stylesheet_link_tag filter	80
1.9.3.4. javascript_include_tag filter	80
1.9.3.5. image_tag filter	80
1.9.3.6. mail_to filter	80

1.9.3.7. html_safe filter	81
1.9.3.8. pluralize filter	81
1.9.3.9. delete_button filter	81
1.9.3.10. delete_button_ajax filter	81
1.9.3.11. update_button filter	81
1.9.3.12. create_button filter	81
1.9.3.13. create_button_ajax filter	81
1.9.3.14. regenerate_oauth_secret_button filter	81
1.9.3.15. link_to filter	81

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our [CTO Chris Wright's message](#).

CHAPTER 1. LIQUID REFERENCE

1.1. WHAT ARE LIQUIDS?

Liquid is a simple programming language used for displaying and processing most of the data from the 3scale system available for API providers. Liquid was originally developed by [Shopify](#) and is used in many other Content Management System (CMS) engines throughout the web. In the 3scale platform, it is used to expose server-side data to your API developers, greatly extending the usefulness of the Developer Portal while maintaining a high level of security.

1.1.1. Pros and cons

Liquids are mainly used to fetch and display server-side data on your Developer Portal pages. However there is much more power in liquids than just this. They serve very well for:

- Altering the DOM and content of a page based on server-side data
- Adding logic to pages, layouts, and partials
- Manipulating the email templates sent to developers

There are some use cases where liquids do not provide the best solution to the problem, mostly situations where you need to use dynamic data such as input from the user or the page URL.

Some general advice is to use them as the primary way to add logic to the page, but then if you find it impossible or overly complicated, switch to JavaScript (or add them to it, as liquids also work well with JS).

1.2. HOW TO USE LIQUIDS

Liquid markup is divided into two types: logic tags and output tags. The logic tags, marked as `{% %}`, are conditional liquid statements that include standard programming language elements such as the *if* clause, loops, and so on.

```
{% 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 %}
```

Output tags, marked as `{{ }}`, are used to display the value of the tag between the curly braces.

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

For documentation on logic tags, please refer to the [Shopify tutorial](#). The full reference of the 3scale liquid output tags can be found both in the [tags](#) section as well as in your Admin Portal under **Help > Liquid Reference**.

1.2.1. Liquid drops, tags, and filters

In the 3scale Developer Portal, you will have access to three types of liquid markup:

- [liquid drops](#) - `{{ current_account.name }}`

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

Liquid drops, tags, and filters are similar to logic and output tags mentioned in [How to use liquids](#).

Liquid drops are the most basic structure, and you will be using them most of the time. They give access to certain values stored in the system such as the name of the user and the identifier of the application. The interpreter handles them in the same way as any other output tags.

On the other hand, liquid tags are a type of logic tag that renders or accesses certain parts of the system for further customization – for example, to render content in the layout or customize email templates.

Filters enable the option of refining results from the drops, converting values, and grouping by some key. There is a group of standard liquid filters, which you can find on the [Shopify website](#) and a group of special 3scale internal filters, which are listed in the [liquid reference](#) section.

1.2.2. The context

The context describes which variables (drops) are available to use on the current page. The basic set includes the following variables:

- `provider`
- `urls`
- `current_user`
- `current_account`
- `today`

These variables are available on every page throughout the Developer Portal, except for email templates. However, most of the built-in pages will have some additional variables available. For example, the edit user form will have a user variable exposed (instead of current user – on edit user page, the identity of the user is already known). To check which variables are available on the current page, there is a special tag: `{% debug:help %}`. It adds the list of all the top-level variables available to use as a comment in the source of the page.

1.2.3. Hierarchy

The direct consequence of the context is that the liquid drops are organized in a hierarchical structure. The available set of variables lists the top level drops that are available. Using them, you can get access to elements much deeper inside the hierarchy. For example, if you would like to show the username of the logged-in user, you would write `{{ current_user.username }}`.

Displaying lower level drops is a little bit more complicated. Assume that you would like to display the name of the only application for a user. Looking at the reference guide, you can see that the method `applications` is a part of the account drop. This means that `applications` is an array of single-application tags. If your users are allowed to have more than one application, then you would have to iterate through them using the logic tags. Otherwise, you can refer to the first (and only) application on the account. The code to display this would look like this: `{{ current_account.applications.first.name }}`.

1.3. LIQUID TEMPLATES

The following variables are available in every Liquid template:

- [provider](#) - all your services, plans and settings under one hood
- [urls](#) - routes to built-in pages of the developers portal - login, signup
- [current_user](#) - username, address and rights of the **currently logged-in user**
- [current_account](#) - messages, applications and plans of the **currently logged-in user**
- [today](#) - current date

Builtin pages can also have other variables available, which are mentioned in the Developer Portal. For example, an edit user form edit will have a **user** variable assigned or while displaying an application detail, you can expect to have a variable **application** accessible.

The type of a variable is important to know to use this reference and can be determined by putting a **{% debug:help %}** tag into the page which will list all the available variables and its types in an HTML comment for you. However, usually they can be guessed quite easily from the method or variable name.

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. FILTERS

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

1.7. DROPS (UP)

1.7.1. Account drop (up)

A developer account. See **User** drop if you are looking for the email addresses or similar information.

```

<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. Methods

1.7.1.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

{{ account.errors.name | inline_errors }}

```

1.7.1.1.2. id

Returns the id of the account

1.7.1.1.3. name

Returns the organization name of the developer's account

1.7.1.1.4. vat_zero_text

Return a text about a vat zero

1.7.1.1.5. vat_rate

Return the vat rate

1.7.1.1.6. unread_messages

Unread messages

1.7.1.1.7. latest_messages

Return the latest messages

1.7.1.1.8. bought_account_plan

Returns the plan the account has contracted

1.7.1.1.9. bought_account_contract

Returns the contract account

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?

Returns whether the account is required to enter credit card details

1.7.1.1.13. credit_card_stored?

Returns whether the account has credit card details stored

1.7.1.1.14. credit_card_missing?

Returns whether the account has no credit card details stored

1.7.1.1.15. timezone

Returns timezone of this account

1.7.1.1.16. paid?

Returns whether the account has at least a paid contract

1.7.1.1.17. on_trial?

Returns whether the account is on trial period, i.e. all his paid contracts has to be in trial period

1.7.1.1.18. telephone_number

Returns the telephone number of the account

1.7.1.1.19. approval_required?

Returns whether the account requires approval?

1.7.1.1.20. created_at

Returns UNIX timestamp of account creation (signup) **Example:** Converting timestamp to JavaScript Date

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

1.7.1.1.21. full_address

Can be composed by legal address, city and state

1.7.1.1.22. applications

Returns the applications of the account

1.7.1.1.23. subscribed_services

Returns a array with ServiceContract drops

1.7.1.1.24. admin

Returns the admin user of this account

1.7.1.1.25. extra_fields_plain_text

Returns the extra fields defined for the account as plain text

1.7.1.1.26. fields_plain_text

Returns the fields defined for the account as plain text

1.7.1.1.27. extra_fields

Returns only extra fields with values of this account **Example:** Print all extra fields

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

1.7.1.1.28. fields

Returns all fields with values of this account **Example:** Print all fields

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

1.7.1.1.29. builtin_fields

1.7.1.1.30. multiple_applications_allowed?

1.7.1.1.31. billing_address

Returns the billing address of this account

1.7.1.1.32. has_billing_address?

Returns whether this account has a billing address or not

1.7.1.1.33. can

Give access to permission methods

```
{% 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)

Example: Using account plan drop in liquid

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

1.7.2.1. Methods

1.7.2.1.1. selected?

Returns whether the plan is selected

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

1.7.2.1.2. bought?

Returns whether the plan is bought

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

1.7.2.1.3. features

Returns an array of available features

1.7.2.1.4. setup_fee

Returns the setup fee

1.7.2.1.5. name

Returns the name of the plan

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

1.7.2.1.6. system_name

Returns the system name of the plan

```
{% 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

Returns the plan id

1.7.2.1.8. free?

The plan is free if it is not **paid**. See the *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?

The plan is **paid** when it has non-zero fixed or setup fee or there are some pricing rules present

```
{% 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?

Returns whether the plan requires approval?

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

1.7.2.1.11. flat_cost

Returns the monthly fixed fee of the plan

1.7.3. Alert drop (up)

Example: Using alert drop in liquid

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

1.7.3.1. Methods

1.7.3.1.1. level

The alert level can be one of 50, 80, 90, 100, 120, 150, 200, 300.

1.7.3.1.2. message

Text message describing the alert, for example *hits per minute: 5 of 5*

1.7.3.1.3. utilization

Decimal number marking the actual utilization that triggered the alert *1.0 is equal to 100%*.

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

1.7.4. Application drop (up)

Example: Using application drop in liquid

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

1.7.4.1. Methods

1.7.4.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.4.1.2. id

Returns the id of the application

1.7.4.1.3. can_change_plan?

Returns **true** if changing of the application is allowed either directly or by request.

1.7.4.1.4. trial?

Returns true if the contract is still in the trial period.

Note: If you change the trial period length of a plan, it does not affect the existing contracts.

1.7.4.1.5. live?

1.7.4.1.6. state

There are three possible states:

- pending
- live
- suspended

1.7.4.1.7. remaining_trial_period_days

Number of day still left in the trial period.

1.7.4.1.8. plan

Returns a plan drop with the plan of the application

1.7.4.1.9. plan_change_permission_name

Returns name of the allowed action

1.7.4.1.10. plan_change_permission_warning

Returns a warning messenger of the allowed action

1.7.4.1.11. contract

1.7.4.1.12. admin_url

Returns the admin_url of the application

1.7.4.1.13. name

Returns the name of the application

1.7.4.1.14. can

1.7.4.1.15. oauth

1.7.4.1.16. pending?

Returns **true** if application state is pending

1.7.4.1.17. buyer_alerts_enabled?

1.7.4.1.18. description

Returns the description of the application

1.7.4.1.19. redirect_url

Returns the redirect url for the OAuth of the application

1.7.4.1.20. filters_limit

Returns the amount of referrer filters allowed for this application

1.7.4.1.21. keys_limit

Returns the amount of application keys allowed for this application

1.7.4.1.22. referrer_filters

Returns the referrer filters associated with this application

1.7.4.1.23. rejection_reason

Returns the reason for rejecting an application

1.7.4.1.24. user_key

Returns the user_key of application

1.7.4.1.25. application_id

Returns the application_id of an application

1.7.4.1.26. key

Returns the application id or the user key

1.7.4.1.27. url

Returns URL of the builtin detail view for this application.

1.7.4.1.28. edit_url

Returns URL of the builtin edit view for this application.

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

Service to which that application belongs to.

1.7.4.1.34. keys

Returns the keys of an application

```
{% 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

Returns non-hidden extra fields with values for this application **Example:** Print all extra fields

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

1.7.4.1.41. fields

Returns all builtin and extra fields with values for this application **Example:** Print all fields

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

1.7.4.1.42. builtin_fields

Returns only builtin fields of the application

1.7.4.1.43. cinstance

1.7.5. ApplicationKey drop [\(up\)](#)

1.7.5.1. Methods

1.7.5.1.1. id

1.7.5.1.2. value

1.7.5.1.3. url

1.7.5.1.4. application

1.7.6. ApplicationPlan drop [\(up\)](#)

1.7.6.1. Methods

1.7.6.1.1. selected?

Returns whether the plan is selected

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

1.7.6.1.2. bought?

Returns whether the plan is bought

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

1.7.6.1.3. features

Returns the visible features of the plan

```
{% 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

Returns the setup fee of the plan

1.7.6.1.5. name

Returns the name of the plan

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

1.7.6.1.6. system_name

Returns the system name of the plan

```
{% 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

Returns the plan id

1.7.6.1.8. free?

The plan is free if it is not **paid**. See the *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?

The plan is **paid** when it has non-zero fixed or setup fee or there are some pricing rules present

```
{% 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?

Returns whether the plan requires approval?

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

1.7.6.1.11. flat_cost

Returns the monthly fixed fee of the plan

1.7.6.1.12. metrics

Returns the metrics of the plan

1.7.6.1.13. usage_limits

Returns the usage limits of the plan

1.7.6.1.14. service

Returns the service of the plan

1.7.7. Base drop (up)

1.7.7.1. Methods

1.7.7.1.1. login_url

1.7.7.1.2. user_identified?

1.7.8. Base drop (up)

1.7.8.1. Methods

1.7.8.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.8.1.2. title

Returns the title result

1.7.8.1.3. kind

Returns the kind of result, can be 'topic' or 'page'

1.7.8.1.4. url

Returns the resource url of the result

1.7.8.1.5. description

Returns a descriptive string for the result

1.7.9. BillingAddressField drop (up)

1.7.9.1. Methods

1.7.9.1.1. input_name

1.7.9.1.2. label

1.7.9.1.3. choices

1.7.9.1.4. errors

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. Methods

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. Methods

1.7.11.1.1. change_plan?

1.7.12. Cas drop (up)

1.7.12.1. Methods

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. Methods

1.7.13.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.13.1.2. id

Returns the id

1.7.13.1.3. can_change_plan?

Returns true if any form of change is possible

1.7.13.1.4. trial?

Returns true if the contract is still in the trial period.

Note: If you change the trial period length of a plan, it does not affect the existing contracts.

1.7.13.1.5. live?

1.7.13.1.6. state

There are three possible states:

- pending
- live
- suspended

1.7.13.1.7. remaining_trial_period_days

Number of day still left in the trial period.

1.7.13.1.8. plan

Returns the plan of the contract

1.7.13.1.9. plan_change_permission_name

Returns name of the allowed action

1.7.13.1.10. plan_change_permission_warning

Returns a warning messenger of the allowed action

1.7.13.1.11. contract

1.7.14. Country drop (up)

1.7.14.1. Methods

1.7.14.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```
{{ 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. Methods

1.7.15.1.1. value

Returns ID of the country

```
{{ account.fields.country.value }} => 42  
compare with:  
{{ account.fields.country }} => 'United States'
```

1.7.15.1.2. name

Returns system name of the field

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. errors

1.7.15.1.10. input_name

1.7.15.1.11. html_id

1.7.15.1.12. label

Returns label of the field

```

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

```

1.7.15.1.13. to_str

Returns name of the country

```

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

```

1.7.15.1.14. choices

1.7.16. CurrentUser drop (up)

1.7.16.1. Methods

1.7.16.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

{{ current_user.errors.name | inline_errors }}

```

1.7.16.1.2. admin?

Returns whether the user is an admin.

```

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

```

1.7.16.1.3. username

Returns the username of the user, html escaped.

1.7.16.1.4. account

Returns the account of the user.

1.7.16.1.5. name

Returns the first and surname of the user.

1.7.16.1.6. email

Returns the email of the user.

1.7.16.1.7. password_required?

This method will return **true** for users using the builtin Developer Portal authentication mechanisms and **false** for those that are authenticated via Janrain, CAS or other single-sign-on method.

```

{{ if user.password_required? }}

{{ endif }}

```

1.7.16.1.8. sections

Returns the list of sections the user has access to.

```

{% 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

Returns the role of the user

1.7.16.1.10. roles_collection

Returns a list of available roles for the user

```

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

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

```

1.7.16.1.11. url

Return the resource url of the user

```

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

```

1.7.16.1.12. edit_url

Return the url to edit the user

```

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

```

1.7.16.1.13. can

Exposes rights of current user which are dependent on your settings and user's role. You can call these methods on the returned object:

- `invite_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`

Returns non-hidden extra fields with values for this user **Example:** Print all extra fields

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

1.7.16.1.15. `fields`

Returns all fields with values for this user **Example:** Print all fields

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

1.7.16.1.16. `builtin_fields`

Returns all builtin fields with values for this user

1.7.17. Error drop (up)

When a form fails to submit because of invalid data, the **errors** array will be available on the related model.

1.7.17.1. Methods

1.7.17.1.1. `attribute`

Returns attribute of the model to this error is related

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

1.7.17.1.2. `message`

Returns description of the error

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

1.7.17.1.3. value

Returns value of the attribute to which the **error** is related

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

1.7.17.1.4. to_str

Returns full description of the error (includes the attribute name)

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

1.7.18. Errors drop (up)

Example: get all errors

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

1.7.18.1. Methods

1.7.18.1.1. empty?

Returns true if there are no errors

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

1.7.18.1.2. present?

Returns true if there are some errors

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

1.7.19. Feature drop (up)

1.7.19.1. Methods

1.7.19.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

{{ feature.errors.name | inline_errors }}

```

1.7.19.1.2. name

Returns the name of the feature

```

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

```

1.7.19.1.3. description

Returns the description of the feature

1.7.19.1.4. has_description?

Returns whether the feature has description

```

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

```

1.7.20. Field drop (up)

1.7.20.1. Methods

1.7.20.1.1. value

Returns value of the field

```

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

```

1.7.20.1.2. name

Returns system name of the field

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. errors

1.7.20.1.10. input_name

Returns name for the HTML input that is expected when the form is submitted.

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

1.7.20.1.11. html_id

Returns a unique field identifier that is commonly used as HTML ID attribute.

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

1.7.20.1.12. label

Returns label of the field

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

1.7.20.1.13. to_str

Returns value of the field if used as variable

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

1.7.20.1.14. choices

Returns array of choices available for that field, if any. For example for a field called **fruit** it may respond with **['apple', 'bannana', 'orange']**.

You can define choices in the 3scale Admin Portal under **Audience > Accounts > Settings > Fields Definitions**. Each of the array elements responds to **id** and **label** which are usually just the same unless the field is a special builtin one (like **country**). It is recommended to use those methods rather than output the **choice** 'as is' for future compatibility.

```
{% 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. Methods

1.7.21.1.1. messages

Return an array of messages

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

1.7.22. Forum drop (up)

1.7.22.1. Methods

1.7.22.1.1. enabled?

Returns **true** if you have forum functionality enabled.

```
{% 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)

Provide useful strings for i18n support.

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

1.7.23.1. Methods

1.7.23.1.1. short_date

Alias for **%b %d**

```
Dec 11
```

1.7.23.1.2. long_date

Alias for **%B %d, %Y**

```
December 11, 2013
```

1.7.23.1.3. default_date

Alias for **%Y-%m-%d**

```
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. Methods

1.7.24.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.24.1.2. email

Returns email address

1.7.24.1.3. accepted?

Returns true if the invitation was accepted

1.7.24.1.4. accepted_at

Returns a date if the invitations was accepted

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

1.7.24.1.5. sent_at

Returns the creation date

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

1.7.24.1.6. resend_url

Returns the url for resend the invitation

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


1.7.24.1.7. url

Returns the resource url

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

1.7.25. Invoice drop (up)

1.7.25.1. Methods

1.7.25.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.25.1.2. friendly_id

Returns a friendly 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

String composed by month and year

1.7.25.1.4. state

1.7.25.1.5. cost

Returns a number with two decimals

```
23.00
```

1.7.25.1.6. currency

1.7.25.1.7. cost_without_vat

Returns cost without VAT

1.7.25.1.8. vat_amount

Returns vat amount

1.7.25.1.9. exists_pdf?

Return true if the pdf was generated

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

Return a AccountDrop

1.7.25.1.18. buyer_account

1.7.25.1.19. line_items

Returns a array of LineltemDrop

```
{% 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

Returns a array of PaymentTransactionDrop

```
{% 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

Return the resource url of the invoice

```

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

```

1.7.25.1.22. pdf_url

Return the resource url of the invoice pdf

```

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

```

1.7.26. Janrain drop [\(up\)](#)

1.7.26.1. Methods

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. LineItem drop [\(up\)](#)

1.7.27.1. Methods

1.7.27.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

{{ 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. Methods

1.7.28.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.28.1.2. id

Returns the id of the message

1.7.28.1.3. subject

If subject is not present then either a truncated body or **(no subject)** string is returned.

1.7.28.1.4. body

Body of the message

1.7.28.1.5. created_at

Returns the creation date

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

1.7.28.1.6. url

URL of the message detail, points either to inbox or outbox.

1.7.28.1.7. state

Either 'read' or 'unread'

1.7.28.1.8. sender

Returns the name of the sender

1.7.28.1.9. to

Returns the name of the receiver

1.7.28.1.10. recipients

1.7.29. Message drop (up)

1.7.29.1. Methods

1.7.29.1.1. type

The possible types of the messages are:

- success (not used by now)
- info
- warning
- danger

1.7.29.1.2. text

1.7.30. Metric drop (up)

1.7.30.1. Methods

1.7.30.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

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

```

1.7.30.1.2. unit

Returns the unit of the metric

```

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

```

1.7.30.1.3. description

Returns the description of the metric

1.7.30.1.4. name

Returns the name of the metric

```

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

```

1.7.30.1.5. system_name

Returns the system name of this metric

```

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

```

1.7.30.1.6. usage_limits

Returns the usage limits of the metric

```

{% 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

Returns the pricing rules of the metric

```

{% 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. Methods

1.7.31.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

{{ page.errors.name | inline_errors }}

```

1.7.31.1.2. title

Returns the title of the page

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

1.7.31.1.3. system_name

Returns system name of the page

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

1.7.32. Page drop (up)

1.7.32.1. Methods

1.7.32.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```
{{ 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. Methods

1.7.33.1.1. page_size

Number of items on one full page.

```
{% 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 %}
  {% endfor %}
```

```

<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

Number of the currently selected page.

1.7.33.1.3. current_offset

Items skipped so far.

1.7.33.1.4. pages

Total number of pages.

1.7.33.1.5. items

Total number of items in all pages together.

1.7.33.1.6. previous

Number of the previous page or empty.

1.7.33.1.7. next

Number of the next page or empty.

1.7.33.1.8. parts

Elements that help to render a user-friendly pagination. See the `[[part-drop[part drop]]` for more information.

1.7.34. Part drop [\(up\)](#)

1.7.34.1. Methods

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?

Returns whether current payment gateway is braintreepayments.com

1.7.35.1.2. authorize_net?

Returns whether current payment gateway is authorize.Net

1.7.35.1.3. type

Returns the type of this payment gateway.

1.7.36. PaymentTransaction drop [\(up\)](#)

1.7.36.1. Methods

1.7.36.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.36.1.2. currency

Returns the currency

```
{% 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

Returns the amount

1.7.36.1.4. created_at

Returns the creation date

1.7.36.1.5. success?

Returns true if was success

1.7.36.1.6. message

Returns the message of the transaction

1.7.36.1.7. reference

Returns the reference

1.7.37. PlanFeature drop (up)

1.7.37.1. Methods

1.7.37.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.37.1.2. name

Returns the name of the feature

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

1.7.37.1.3. description

Returns the description of the feature

1.7.37.1.4. has_description?

Returns whether the feature has description

```
{% 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

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.38.1.2. body

Text of the post.

1.7.38.1.3. topic

Every post belongs to a `[[topic-drop[topic]`

1.7.38.1.4. created_at

Date when this post created

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

1.7.38.1.5. url

The url of this post within its topic

1.7.39. Post drop (up)

1.7.39.1. Methods

1.7.39.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```
{{ 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

Returns the cost per unit of the pricing rule **Example:** Using pricing rule drop in liquid

```
<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. min

Returns the minimum value of the pricing rule

1.7.40.1.3. max

Returns the maximum value of the pricing rule

1.7.40.1.4. plan

Returns plan of pricing rule

1.7.41. Provider drop (up)

1.7.41.1. Methods

1.7.41.1.1. name

Returns the name of your organization.

```
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

Returns the payment gateway associated with your organization

1.7.41.1.3. domain

Domain of your Developer Portal

1.7.41.1.4. timezone

Returns the timezone that you use. You can change your timezone in the Admin Portal. Select **Dashboard > Account Settings**. In the **Overview** page, scroll down to **Account Details**, click **Edit** and modify the **Time Zone** field.

1.7.41.1.5. support_email

Support email of the account

1.7.41.1.6. finance_support_email

Finance support email of the account

1.7.41.1.7. telephone_number

Returns the telephone number of the account

1.7.41.1.8. multiple_applications_allowed?

True if developers can have more separate applications with their own keys, stats, and so on. This depends on your 3scale plan.

```

{% 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

Return the logo url

1.7.41.1.10. multiple_services_allowed?

True if your 3scale plan allows you to manage multiple APIs as separate services.

```
{% 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. multiple_users_allowed?

True if the developer accounts can have multiple logins associated with them. This depends on your 3scale plan and if its visibility has been turned on for your Developer Portal in the Admin Portal under **Audience > Developer Portal > Feature Visibility**

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

1.7.41.1.13. account_plans

Returns all published account plans.

```
<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

Returns all defined services.

```
<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 %}
```

1.7.41.15. signups_enabled?

You can enable or disable signups in the 3scale Admin Portal under **Audience > Accounts > Settings > Usage Rules**.

1.7.41.16. account_management_enabled?

You can enable or disable account management in the 3scale Admin Portal under **Audience > Accounts > Settings > Usage Rules**.

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)

Example: Using request drop in liquid

```
<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

Returns the URI of the request

1.7.43.1.2. host_with_port

Returns the host with port of the request

1.7.43.1.3. host

Returns the host part of the request URL

1.7.43.1.4. path

Returns the path part of the request URL

```
{% 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

Return internal name of the role, important for the system

1.7.44.1.2. description

Return a descriptive text for the role

1.7.45. Search drop (up)

1.7.45.1. Methods

1.7.45.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.45.1.2. query

Returns the searched string

```
<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

Returns the number of matching elements

1.7.45.1.4. results

Returns an array of results for que search

1.7.46. Service drop (up)

1.7.46.1. Methods

1.7.46.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.46.1.2. name

Returns the name of the service

1.7.46.1.3. system_name

Returns the system name of the service

```
{% case service.system_name %}
{% when 'api' %}
  API is our newest service!
{% when 'old' %}
  Unfortunately we dont allow more signups to our old service.
{% endcase %}
```

1.7.46.1.4. description

Returns the description of the service

1.7.46.1.5. subscribed?

Returns whether the service is subscribed

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

1.7.46.1.6. subscription

Returns a subscription(**ServiceContract** drop) if the currently logged in user is subscribed to this service, Nil otherwise.

```
{% 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

Returns the **published** application plans of the service

```
{% 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

Returns the *published* service plans of the service

```
<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

Returns the application plans of the service

1.7.46.1.12. features

Returns the visible features of the service

```
{% if service.features.size > 0 %}
  <p>{{ service.name }} has following features:</p>
  <ul>
    {% for feature in service.features %}
      <li>{{ feature.name }}</li>
    {% endfor %}
  </ul>
```

```

</ul>
{% else %}
  <p>Unfortunately, {{ service.name }} currently has no features.</p>
{% endif %}

```

1.7.46.113. apps_identifier

Depending on the authentication mode set, returns either 'ID', 'API key' or 'Client ID' for OAuth authentication.

```

{{ service.application_key_name }}

```

1.7.46.114. backend_version

1.7.46.115. referrer_filters_required?

1.7.46.116. metrics

Returns the metrics of the service

```

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

```

1.7.46.117. support_email

Support email of the service

1.7.47. ServiceContract drop (up)

1.7.47.1. Methods

1.7.47.1.1. errors

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```

{{ service_contract.errors.name | inline_errors }}

```

1.7.47.1.2. id

1.7.47.1.3. can_change_plan?

Returns true if any form of change is possible

1.7.47.1.4. trial?

Returns true if the contract is still in the trial period.

Note: If you change the trial period length of a plan, it does not affect the existing contracts.

1.7.47.1.5. live?

1.7.47.1.6. state

There are three possible states:

- pending
- live
- suspended

1.7.47.1.7. remaining_trial_period_days

Number of day still left in the trial period.

1.7.47.1.8. plan

Returns the plan of the contract

1.7.47.1.9. plan_change_permission_name

Returns name of the allowed action

1.7.47.1.10. plan_change_permission_warning

Returns a warning messenger of the allowed action

1.7.47.1.11. contract

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. applications

1.7.47.1.17. can

Exposes specific rights of the current user for that subscription.

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

1.7.48. ServicePlan drop [\(up\)](#)

1.7.48.1. Methods

1.7.48.1.1. selected?

Returns whether the plan is selected

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

1.7.48.1.2. bought?

Returns whether the plan is bought

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

1.7.48.1.3. features

Returns the visible features of the plan

```
{% 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

Returns the setup fee of the plan

1.7.48.1.5. name

Returns the name of the plan

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

1.7.48.1.6. system_name

Returns the system name of the plan

```
{% 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

Returns the plan id

1.7.48.1.8. free?

The plan is free if it is not **paid**. See the *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?

The plan is **paid** when it has non-zero fixed or setup fee or there are some pricing rules present

```
{% 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?

Returns whether the plan requires approval?

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

1.7.48.1.11. flat_cost

Returns the monthly fixed fee of the plan

1.7.48.1.12. service

Example: Using service plan drop in liquid

```
<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

Returns current month (1-12)

1.7.50.1.2. day

Returns current day of the month (1-31)

1.7.50.1.3. year

Returns current year **Example:** Create dynamic copyright

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

1.7.50.1.4. beginning_of_month

Returns date of beginning of current month

```
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

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

```
{{ 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

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.52.1.2. title

Name of the topic. Submitted when first post to the thread is posted.

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?

True if the path of the current page is the same as the one for the URL or it is a *direct subpath* of it, that is, extended by ID. For example with `{{ urls.outbox }}` these will return true:

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

But not these:

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

See also '#active?', '#current?'.

1.7.53.1.5. current?

True if the URL path is the the same as of the current. Parameters and other components are not taken into account. See also '#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?

True if the current page is in the same menu structure as this URL. See also `#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

URL of a signup page. Accessible for all.

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

1.7.54.1.5. search

URL to which all the search requests should be sent

```
<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

URL to a service subscription page. Only for logged in users.

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

1.7.54.1.10. compose_message

URL to a page that allows the developer contact provider via the internal messaging system.

1.7.54.1.11. messages_outbox

URL to the list of messages sent by a developer.

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

URL or Nil if user account management is disabled. Check **Usage Rules** in the 3scale Admin Portal under **Audience > Accounts > Settings > Usage Rules**

1.7.54.1.19. access_details

A page with API key(s) and other authentication info. Differs depending on the authentication strategy.

1.7.54.1.20. payment_details

1.7.54.1.21. new_invitation

Page to invite new users

1.7.54.1.22. invitations

List of all the sent invitations

1.7.54.1.23. dashboard

1.7.54.1.24. applications

1.7.54.1.25. api_access_details

1.7.54.1.26. services

1.7.54.1.27. messages_inbox

URL to the list of received messages.

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)

Example: Using usage limit drop in liquid

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

1.7.55.1. Methods

1.7.55.1.1. period

Returns the period of the usage limit

1.7.55.1.2. metric

Usually **hits** but can be any custom method.

1.7.55.1.3. value

Returns the value of the usage limit

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

If a form of this model is rendered after an unsuccessful submit, this returns errors that occurred.

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

1.7.56.1.2. admin?

Returns whether the user is an admin.

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

1.7.56.1.3. username

Returns the username of the user, html escaped.

1.7.56.1.4. account

Returns the account of the user.

1.7.56.1.5. name

Returns the first and surname of the user.

1.7.56.1.6. email

Returns the email of the user.

1.7.56.1.7. password_required?

This method will return **true** for users using the builtin Developer Portal authentication mechanisms and **false** for those that are authenticated via Janrain, CAS or other single-sign-on method.

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

1.7.56.1.8. sections

Returns the list of sections the user has access to.

```
{% 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

Returns the role of the user

1.7.56.1.10. roles_collection

Returns a list of available roles for the user

```
{% 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

Return the resource url of the user

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

1.7.56.1.12. edit_url

Return the url to edit the user

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

1.7.56.1.13. can

Give access to permission methods

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

1.7.56.1.14. extra_fields

Returns non-hidden extra fields with values for this user **Example:** Print all extra fields

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

1.7.56.1.15. fields

Returns all fields with values for this user **Example:** Print all fields

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

1.7.56.1.16. builtin_fields

Returns all builtin fields with values for this user

1.8. TAGS (UP)

1.8.1. Tag 'braintree_customer_form' (up)

Renders a form to enter data required for Braintree Blue payment gateway

1.8.2. Tag 'csrf' (up)

Renders the cross site request forgery meta tags.

Example: Using csrf tag in liquid

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

1.8.3. Tag 'content' (up)

Renders body of a page. Use this only inside a layout.

1.8.4. Tag 'content_for' (up)

1.8.5. Tag 'debug' (up)

Prints all liquid variables available in a template into an HTML comment.' We recommend **to remove this tag** from public templates.

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

1.8.6. Tag 'email' (up)

The **email** tag allows you to customize headers of your outgoing emails and is available only inside the email templates.

There are several convenience subtags such as **cc** or **subject** (see the table below) that simplify the job but you can also use a **header** subtag to set an arbitrary SMTP header for the message.

Subtag	Description	Example
subject	dynamic subject	{% subject = 'Greetings from Example company!' %}
cc	carbon copy	{% cc = 'boss@example.com' %}
bcc	blind carbon copy	{% bcc = 'all@example.com' %}

from	the actual sender	{% from = 'system@example.com' %}
reply-to		{% reply-to = 'support@example.com' %}
header	custom SMTP header	{% header 'X-SMTP-Group' = 'Important' %}
do_not_send	discard the email	{% do_not_send %}

Example: Conditional blind carbon copy

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

Example: Disabling emails at all

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

Example: 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)

Renders informational or error messages of the system.

DEPRECATED: This tag is deprecated, use FlashDrop instead.

Example: Using flash tag in liquid

```
{% flash %}
```

1.8.8. Tag 'footer' (up)

Renders a footer html snippet.

DEPRECATED: This tag is deprecated. Use a Developer Portal partial instead.

1.8.9. Tag 'form' (up)

Renders a form tag with an action and class attribute specified, depending on the name of the form. The supported forms are:

Form	Allowed Field Names	Spam Protection	Notes
application.create	<ul style="list-style-type: none">● application[name]● application[description]● application[<any-extra-field>]	No	
application.update	<ul style="list-style-type: none">● application[name]● application[description]● application[<any-extra-field>]	No	

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>] 	Yes	<p>Sign Up directly to plans of your choice by adding one or more hidden fields with a name plan_ids[]. If a parameter of such name is found in the current URL, the input field is filled automatically.</p>
--------	---	-----	--

Example: A form to create an application

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

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

```
{% endform %}
```

1.8.10. Tag 'latest_forum_posts' (up)

An HTML table with latest forum posts.

DEPRECATED: Use **forum** drop instead.

Example: Using latest_forum_posts tag liquid

```
{% latest_forum_posts %}
```

1.8.11. Tag 'latest_messages' (up)

Renders a html snippet with the latest messages for the user.

Example: Using latest_messages tag liquid

```
{% latest_messages %}
```

1.8.12. Tag 'logo' (up)

Renders the logo.

DEPRECATED: This tag is deprecated, use `{{ provider.logo_url }}` instead.

Example: Using menu tag in liquid

```
{% logo %}
```

1.8.13. Tag 'menu' (up)

DEPRECATED: This tag is deprecated, use `'{% include "menu" %}'` instead.

1.8.14. Tag 'oldfooter' (up)

Renders a footer html snippet.

DEPRECATED: This tag is deprecated. Use a Developer Portal partial instead.

1.8.15. Tag 'plan_widget' (up)

Includes a widget to review or change application plan

```
{% 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)

This tag includes portlet by system name.

1.8.17. Tag 'submenu' (up)

Renders a submenu html snippet for a logged in user.

DEPRECATED: This tag is deprecated, use a 'submenu' partial instead

Example: Using submenu tag in liquid

```
{% submenu %}
```

1.8.18. Tag '3scale_essentials' (up)

1.8.19. Tag 'user_widget' (up)

Renders a user widget html snippet.

DEPRECATED: This tag is deprecated. Use a Developer Portal partial instead.

Example: Using user_widget tag in liquid

```

{% 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

Outputs error class if argument is not empty **Example:** Using error_class to show output an error class

```
■
```

1.9.1.2. inline_errors filter

Outputs error fields inline in paragraph **Example:** Using inline_errors to show errors inline

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

1.9.2. ParamFilter filters (up)

1.9.2.1. to_param filter

Converts a supplied drop to URL parameter if possible. **Example:** Using to_param filter in liquid

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

1.9.3. Common filters (up)

1.9.3.1. group_by filter

Group collection by some key **Example:** Group applications by service

```
{% 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. any filter

True if any string in the collection equals to the parameter **Example:** Are there any pending applications of the current account?

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

1.9.3.3. stylesheet_link_tag filter

Stylesheet link

1.9.3.4. javascript_include_tag filter

Javascript include tag

1.9.3.5. image_tag filter

Outputs an tag using the parameters as its **src** attribute.

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

1.9.3.6. mail_to filter

Converts email address to a 'mailto' link.

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

1.9.3.7. `html_safe` filter

Marks content as HTML safe so that it is not escaped.

1.9.3.8. `pluralize` filter

Convert word to plural form

1.9.3.9. `delete_button` filter

Generates a button to delete a resource present on the URL. First parameter is a URL, second is a title. You can also add more HTML tag attributes as a third parameter.

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

1.9.3.10. `delete_button_ajax` filter

Generates a button to delete a resource present on the URL using AJAX. First parameter is a URL, second is a title.

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

1.9.3.11. `update_button` filter

Generates a button to 'update' (HTTP PUT request) a resource present on the URL. First parameter is a URL, second is a title. You can also add more HTML tag attributes as a third parameter.

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

1.9.3.12. `create_button` filter

Generates a button to create a resource present on the URL. First parameter is a URL, second is a title.

```
{{ '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

Create link from given text

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