



# Red Hat 3scale API Management 2.8

## Glossary

Learn about the terms used in Red Hat 3scale API Management.



## Red Hat 3scale API Management 2.8 Glossary

---

Learn about the terms used in Red Hat 3scale API Management.

## Legal Notice

Copyright © 2020 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## Abstract

This guide documents the terms used with Red Hat 3scale API Management 2.8.

## Table of Contents

<b>CHAPTER 1. 3SCALE GLOSSARY</b> .....	<b>4</b>
1.1. SPECIAL CHARACTERS	4
1.1.1. [Your_admin_domain]	4
1.1.2. [Your_API_name]	4
1.1.3. [Your_API_service]	4
1.1.4. [Your_product_name]	4
1.1.5. 3scale API docs	4
1.1.6. 3scale backend	4
1.2. A	5
1.2.1. account	5
1.2.2. ActiveDocs	5
1.2.3. access token	5
1.2.4. Admin Portal	5
1.2.5. Administration Portal	5
1.2.6. API	5
1.2.7. APIsonator	5
1.2.8. API backend	5
1.2.9. API consumer	5
1.2.10. API credential	5
1.2.11. API endpoint	6
1.2.12. API provider	6
1.2.13. API key	6
1.2.14. API manager	6
1.2.15. API method	6
1.2.16. API product	6
1.2.17. APIcast	6
1.2.18. APIcast gateway	6
1.2.19. app identifier	6
1.2.20. app key	6
1.2.21. application	6
1.2.22. application programming interface	7
1.2.23. authentication	7
1.3. B	7
1.3.1. backend	7
1.3.2. base URL	7
1.4. C	7
1.4.1. client	7
1.4.2. CMS	7
1.4.3. code plugin	7
1.5. D	8
1.5.1. deprecated	8
1.5.2. developer account	8
1.5.3. Developer Portal	8
1.5.4. discontinued	8
1.6. E	8
1.6.1. endpoint	8
1.6.2. end user	8
1.7. F	8
1.7.1. field definition	8
1.8. G	8
1.9. H	8

1.9.1. hit	9
1.9.2. host URL	9
1.10. I	9
1.10.1. integration error	9
1.11. J	9
1.12. K	9
1.13. L	9
1.14. M	9
1.14.1. management portal	9
1.14.2. mapping rule	9
1.14.3. method	9
1.14.4. metric	9
1.15. N	10
1.16. O	10
1.16.1. OAS	10
1.16.2. OpenAPI Specification	10
1.17. P	10
1.17.1. plan	10
1.17.2. potential upgrade	10
1.17.3. plugin	10
1.17.4. product	10
1.18. Q	10
1.19. R	10
1.19.1. removed	10
1.20. S	11
1.20.1. service	11
1.20.2. service token	11
1.21. T	11
1.21.1. tenant account	11
1.22. U	11
1.22.1. user key	11
1.23. V	11
1.24. W	11
1.24.1. wrapper	11
1.25. X	11
1.26. Y	11
1.26.1. your domain	11
1.27. Z	11



## CHAPTER 1. 3SCALE GLOSSARY

This section acts as a glossary of common terms used within the 3scale platform. Following is the structure of glossary entries:

- A glossary term
- One or more of each or both of the following items:
  - A definition of the term, optionally followed by one or two explanatory sentences and a *See also* reference that points to a related term or terms. Example:

**hit:** The built-in metric to which all methods report. Additional top-level metrics can be added here in order to track other usage that should not increase the hit count. See also [Section 1.14.4, "metric"](#).

In this case, **hit** is related to **metric** because hit is a built-in metric.

- A *See* reference that points to a preferred synonym for that term or to the spelled-out version of an abbreviated term. Example:

**account:** See [Section 1.21.1, "tenant account"](#), [Section 1.5.2, "developer account"](#).

In this case, the reader is redirected from **account** to either of the relevant terms containing qualifiers.

- If there is more than one definition for a term, each item is numbered.

### 1.1. SPECIAL CHARACTERS

#### 1.1.1. [Your\_admin\_domain]

Admin domain created by the user.

#### 1.1.2. [Your\_API\_name]

API created by the user. See also [Section 1.2.22, "application programming interface"](#).

#### 1.1.3. [Your\_API\_service]

See [Section 1.1.2, "\[Your\\_API\\_name\]"](#).

#### 1.1.4. [Your\_product\_name]

Product created by the user. See also [Section 1.17.4, "product"](#).

#### 1.1.5. 3scale API docs

Based on Open API Specification (OAS), a 3scale specification for documenting REST APIs from Red Hat 3scale API Management. See also [Section 1.2.2, "ActiveDocs"](#), [Section 1.16.2, "OpenAPI Specification"](#).

#### 1.1.6. 3scale backend

Part of the 3scale API Manager, along with System.



## 1.2. A

### 1.2.1. account

See [Section 1.21.1, "tenant account"](#) , [Section 1.5.2, "developer account"](#) .

### 1.2.2. ActiveDocs

Based on Open API Specification (OAS), a 3scale specification for documenting REST APIs created by users. APIs get interactive documentation which makes it easier for developers to understand APIs and also to test them without any installation. See also [Section 1.1.5, "3scale API docs"](#) , [Section 1.16.2, "OpenAPI Specification"](#) .

### 1.2.3. access token

Personal token that allow users authenticate against the Account Management API, the Analytics API and the Billing API through HTTP Basic Auth. Users can create multiple access tokens with custom scopes and permissions. See also [Section 1.1.5, "3scale API docs"](#) , [Section 1.20.2, "service token"](#) .

### 1.2.4. Admin Portal

The central portal for API providers to manage and secure access to their Products. From this portal, API providers can configure the integration of their Product with 3scale, manage their application plans, give access to internal members and external customers, limit traffic per application keys, and customize their Developer Portal. See also [Section 1.5.3, "Developer Portal"](#) , [Section 1.17.4, "product"](#) .

### 1.2.5. Administration Portal

See [Section 1.2.4, "Admin Portal"](#) .

### 1.2.6. API

See [Section 1.2.22, "application programming interface"](#) .

### 1.2.7. APIsonator

See [Section 1.1.6, "3scale backend"](#) .

### 1.2.8. API backend

Implementation of an API deployed in a host. See also [Section 1.2.22, "application programming interface"](#) , [Section 1.3.1, "backend"](#) .

### 1.2.9. API consumer

An individual, group or company that accesses an API or a product managed by the API provider. The term may refer both to the organization and the software applications written by the organizations to use the API. A given organization may have one or more applications which access the API. See also [Section 1.2.12, "API provider"](#) .

### 1.2.10. API credential

See [Section 1.2.13, "API key"](#), [Section 1.22.1, "user key"](#).

### 1.2.11. API endpoint

See [Section 1.2.22, "application programming interface"](#), [Section 1.14.3, "method"](#).

### 1.2.12. API provider

The entity that owns the APIs and products, and gives access to them by using Red Hat 3scale API Management. An API provider may make their APIs accessible to other teams internally within their organization or externally to third-party developers, partners, or the public. It might include one or multiple tenants. See also [Section 1.2.9, "API consumer"](#), [Section 1.21.1, "tenant account"](#).

### 1.2.13. API key

A type of credential for an application to be allowed to make calls on a specific API or product. API Keys are a specific type of authentication pattern. See also [Section 1.2.22, "application programming interface"](#).

### 1.2.14. API manager

See [Section 1.2.4, "Admin Portal"](#).

### 1.2.15. API method

See [Section 1.14.3, "method"](#).

### 1.2.16. API product

See [Section 1.17.4, "product"](#).

### 1.2.17. APIcast

An API gateway based on NGINX used to integrate internal and external APIs with Red Hat 3scale API Management. APIcast is the interface that handles API requests, and depending on its configuration it can handle access control, rate limiting, security filtering, logging, routing, caching, etc. See also [Section 1.2.22, "application programming interface"](#), [Section 1.17.4, "product"](#).

### 1.2.18. APIcast gateway

See [Section 1.2.17, "APIcast"](#).

### 1.2.19. app identifier

Authentication method working with app identifier. See also [Section 1.2.20, "app key"](#).

### 1.2.20. app key

Authentication method with app keys. See also [Section 1.2.19, "app identifier"](#).

### 1.2.21. application

A piece of software code which is developed to perform some logic. The application will normally have associated to it within the 3scale system a unique set of credentials for the API, a traffic history of the calls sent to the API and meta-data captured at application creation. Applications are linked to developer accounts. See [Section 1.2.9, "API consumer"](#). See also [Section 1.5.2, "developer account"](#).

### 1.2.22. application programming interface

1. An interface to a software component that can be invoked at a distance over a communications network using standards based technologies.
2. A logical bundle of processes, functions and methods offered by a programming library, as an abstraction layer, to be used by another computer program.

See also [Section 1.1.2, "\[Your\\_API\\_name\]"](#), [Section 1.2.17, "APIcast"](#), [Section 1.2.13, "API key"](#), [Section 1.3.1, "backend"](#), [Section 1.6.1, "endpoint"](#), [Section 1.6.2, "end user"](#), [Section 1.14.3, "method"](#), [Section 1.17.4, "product"](#), [Section 1.20.1, "service"](#).

### 1.2.23. authentication

The process of verifying the identity of a user or server requesting access to an API or product.

## 1.3. B

### 1.3.1. backend

1. In the context of API as a Product, one or more APIs bundled in a product. See also [Section 1.3.2, "base URL"](#), [Section 1.17.4, "product"](#).
2. As an API, see also [Section 1.2.8, "API backend"](#).
3. As a 3scale component, see [Section 1.1.6, "3scale backend"](#).

### 1.3.2. base URL

1. In the context of APIs as a Product, the private endpoint defined in the backend. See also [Section 1.3.1, "backend"](#), [Section 1.6.1, "endpoint"](#).
2. The URL address defined in the Integration Settings for the API Gateway. These addresses are the Staging Public Base URL, and the Production Public Base URL.

## 1.4. C

### 1.4.1. client

See [Section 1.2.9, "API consumer"](#).

### 1.4.2. CMS

See [Section 1.5.3, "Developer Portal"](#).

### 1.4.3. code plugin

See [Section 1.17.3, "plugin"](#).

## 1.5. D

### 1.5.1. deprecated

Pertaining to an entity, such as a feature or supported configuration, that is supported but no longer recommended and that might become obsolete. See also [Section 1.19.1, "removed"](#).

### 1.5.2. developer account

1. In 3scale Hosted, all accounts the API provider sees are developer accounts.
2. In 3scale On-premises, in the tenant Admin Portal, accounts are considered as developer accounts. See also [Section 1.2.21, "application"](#), [Section 1.21.1, "tenant account"](#).

### 1.5.3. Developer Portal

The site where developers can subscribe to APIs. From this site, developers can manage their subscription, have access to their APIs, create applications, access the interactive API documentation, 3scale API docs, see their API consumption, etc. See also [Section 1.2.4, "Admin Portal"](#).

### 1.5.4. discontinued

See [Section 1.19.1, "removed"](#).

## 1.6. E

### 1.6.1. endpoint

One end of a communication channel. When an API interacts with another system, the touchpoints of this communication are considered endpoints. See also [Section 1.2.22, "application programming interface"](#), [Section 1.3.2, "base URL"](#), [Section 1.14.2, "mapping rule"](#), [Section 1.14.3, "method"](#).

### 1.6.2. end user

The user of an application which calls one or multiple products and APIs. See also [Section 1.2.22, "application programming interface"](#), [Section 1.17.4, "product"](#).

## 1.7. F

### 1.7.1. field definition

Fields displayed when either a new user, application or account is created.

## 1.8. G

## 1.9. H

## 1.9.1. hit

The built-in metric to which all methods report. Additional top-level metrics can be added here in order to track other usage that should not increase the hit count. See also [Section 1.14.4, "metric"](#).

## 1.9.2. host URL

See [Section 1.3.2, "base URL"](#).

## 1.10. I

### 1.10.1. integration error

Error messages generated by 3scale when calls are performed with an incorrect set of credentials, incorrect IDs, URL addresses, etc. These messages can be errors coming from the application which is calling the API, or they can be a configuration error with the integration of the API with 3scale.

## 1.11. J

## 1.12. K

## 1.13. L

## 1.14. M

### 1.14.1. management portal

See [Section 1.2.4, "Admin Portal"](#).

### 1.14.2. mapping rule

Rule that associates incoming calls from specific endpoints to the corresponding methods and metrics created in 3scale. Usage tracking, endpoint access and limits are based on the methods and metrics that are configured with mapping rules. See also [Section 1.6.1, "endpoint"](#), [Section 1.14.3, "method"](#), [Section 1.14.4, "metric"](#).

### 1.14.3. method

Allowed interactions -such as GET, POST or DELETE- with an API or a product. Methods are used to track the usage of products and API on 3scale. A method can be added for each of the HTTP methods available on the API endpoints for the API. By default on 3scale, method calls trigger the built-in hits metric. See also [Section 1.2.22, "application programming interface"](#), [Section 1.6.1, "endpoint"](#), [Section 1.14.2, "mapping rule"](#), [Section 1.14.4, "metric"](#).

### 1.14.4. metric

Track of specific calls to an API. This tracking is done by mapping the calls to one or more URL patterns in the Mapping Rules section of the integration page. Metrics are cumulative and not discrete. The built-in top level metric on 3scale is Hits. Additional top level metrics can be added if needed, and these are

considered separately. See also [Section 1.9.1, "hit"](#), [Section 1.14.2, "mapping rule"](#), [Section 1.14.3, "method"](#).

## 1.15. N

## 1.16. O

### 1.16.1. OAS

See [Section 1.16.2, "OpenAPI Specification"](#)

### 1.16.2. OpenAPI Specification

A standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. See also [Section 1.1.5, "3scale API docs"](#), [Section 1.2.2, "ActiveDocs"](#).

## 1.17. P

### 1.17.1. plan

Approach for granting access to specific APIs and endpoints, limiting traffic and monetizing API usage. 3scale has different types of plans that can be used on their own or in conjunction, such as Application plans and Account plans.

### 1.17.2. potential upgrade

Developer accounts that trigger traffic limit alerts. These developer accounts could potentially be upgraded to a plan which meets their needs in terms of traffic volume.

### 1.17.3. plugin

A wrapper for the 3scale API that enables API Access Control and API Traffic Reporting without having to run another server. A plugin lives inside the code that powers the API. The plugin calls the 3scale API for access control and traffic reporting.

### 1.17.4. product

A bundle of one or more API backends. See also [Section 1.1.4, "\[Your\\_product\\_name\]"](#), [Section 1.2.4, "Admin Portal"](#), [Section 1.2.17, "APIcast"](#), [Section 1.2.22, "application programming interface"](#), [Section 1.3.1, "backend"](#), [Section 1.6.2, "end user"](#).

## 1.18. Q

## 1.19. R

### 1.19.1. removed

Pertaining to an entity, such as a feature or supported configuration, that is not available in the product and is no longer supported. See also [Section 1.5.1, "deprecated"](#).

## 1.20. S

### 1.20.1. service

See [Section 1.2.22, "application programming interface"](#).

### 1.20.2. service token

Key used to authenticate against the Service Management API. These keys are auto generated, unique per service and shared between the users of an account. Use Service Tokens from within the 3scale API docs. See also [Section 1.2.3, "access token"](#).

## 1.21. T

### 1.21.1. tenant account

In 3scale on-premises, the Master Admin Portal considers accounts as tenant accounts. See also [Section 1.2.12, "API provider"](#), [Section 1.5.2, "developer account"](#).

## 1.22. U

### 1.22.1. user key

See [Section 1.2.20, "app key"](#).

## 1.23. V

## 1.24. W

### 1.24.1. wrapper

See [Section 1.17.3, "plugin"](#).

## 1.25. X

## 1.26. Y

### 1.26.1. your domain

See [Section 1.1.1, "\[Your\\_admin\\_domain\]"](#).

## 1.27. Z

