networking.istio.io

destinationrules gateways sidecars

virtualservices

Type

`object`

Specification

Property	Type	Description
`spec`	`object`	Configuration affecting load balancing, outlier detection, etc. See more details at: https://istio.io/docs/reference/config/networking/destination-rule.html ↗
`status`	`object`	

.spec

Description

Configuration affecting load balancing, outlier detection, etc. See more details at: https://istio.io/docs/reference/config/networking/destination-rule.html

Type

`object`

Property	Type	Description
`exportTo`	`array`	A list of namespaces to which this destination rule is exported.
`host`	`string`	The name of a service from the service registry.
`subsets`	`array`	
`trafficPolicy`	`object`	
`workloadSelector`	`object`	

.spec.exportTo

Description

A list of namespaces to which this destination rule is exported.

Type

`array`

.spec.exportTo[]

Type

`string`

.spec.subsets

Type

`array`

.spec.subsets[]

Type

`object`

Property	Type	Description
`labels`	`object`	
`name`	`string`	Name of the subset.
`trafficPolicy`	`object`	Traffic policies that apply to this subset.

.spec.subsets[].labels

Type

`object`

.spec.subsets[].trafficPolicy

Description

Traffic policies that apply to this subset.

Type

`object`

Property	Type	Description
`connectionPool`	`object`	
`loadBalancer`	`object`	Settings controlling the load balancer algorithms.
`outlierDetection`	`object`	

Property	Type	Description
`portLevelSettings`	`array`	Traffic policies specific to individual ports.
`tls`	`object`	TLS related settings for connections to the upstream service.
`tunnel`	`object`	

.spec.subsets[].trafficPolicy.connectionPool

Type

`object`

Property	Type	Description
`http`	`object`	HTTP connection pool settings.
`tcp`	`object`	Settings common to both HTTP and TCP upstream connections.

.spec.subsets[].trafficPolicy.connectionPool.http

Description

HTTP connection pool settings.

Type

`object`

Property	Type	Description
`h2UpgradePolicy`	`string`	Specify if http1.1 connection should be upgraded to http2 for the associated destination.
`http1MaxPendingRequests`	`integer`	
`http2MaxRequests`	`integer`	Maximum number of active requests to a destination.
`idleTimeout`	`string`	The idle timeout for upstream connection pool connections.
`maxRequestsPerConnection`	`integer`	Maximum number of requests per connection to a backend.
`maxRetries`	`integer`	
`useClientProtocol`	`boolean`	If set to true, client protocol will be preserved while initiating connection to backend.

.spec.subsets[].trafficPolicy.connectionPool.tcp

Description

Settings common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
`connectTimeout`	`string`	TCP connection timeout.
`maxConnectionDuration`	`string`	The maximum duration of a connection.
`maxConnections`	`integer`	Maximum number of HTTP1 /TCP connections to a destination host.
`tcpKeepalive`	`object`	If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

.spec.subsets[].trafficPolicy.connectionPool.tcp.tcpKeepalive

Description

If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

Type

`object`

Property	Type	Description
`interval`	`string`	The time duration between keep-alive probes.
`probes`	`integer`	
`time`	`string`	

.spec.subsets[].trafficPolicy.loadBalancer

Description

Settings controlling the load balancer algorithms.

Type

`object`

Property	Type	Description
`consistentHash`	`object`	
`localityLbSetting`	`object`	
`simple`	`string`	
`warmupDurationSecs`	`string`	Represents the warmup duration of Service.

.spec.subsets[].trafficPolicy.loadBalancer.consistentHash

Type

`object`

Property	Type	Description
`httpCookie`	`object`	Hash based on HTTP cookie.
`httpHeaderName`	`string`	Hash based on a specific HTTP header.
`httpQueryParameterName`	`string`	Hash based on a specific HTTP query parameter.

Property	Type	Description
`maglev`	`object`	The Maglev load balancer implements consistent hashing to backend hosts.
`minimumRingSize`	`integer`	Deprecated.
`ringHash`	`object`	The ring/modulo hash load balancer implements consistent hashing to backend hosts.
`useSourceIp`	`boolean`	Hash based on the source IP address.

.spec.subsets[].trafficPolicy.loadBalancer.consistentHash.httpCookie

Description

Hash based on HTTP cookie.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the cookie.
`path`	`string`	Path to set for the cookie.

Property	Type	Description
`ttl`	`string`	Lifetime of the cookie.

.spec.subsets[].trafficPolicy.loadBalancer.consistentHash.maglev

Description

The Maglev load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`tableSize`	`integer`	The table size for Maglev hashing.

.spec.subsets[].trafficPolicy.loadBalancer.consistentHash.ringHash

Description

The ring/modulo hash load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`minimumRingSize`	`integer`	

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting

Type

`object`

Property	Type	Description
`distribute`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`enabled`	`boolean`	enable locality load balancing, this is DestinationRule-level and will override mesh wide settings in entirety.
`failover`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`failoverPriority`	`array`	failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.distribute

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.distribute[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating locality, '/' separated, e.g.
`to`	`object`	Map of upstream localities to traffic distribution weights.

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.distribute[].to

Description

Map of upstream localities to traffic distribution weights.

Type

`object`

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.failover

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.failover[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating region.
`to`	`string`	

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.failoverPriority

Description

failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

Type

`array`

.spec.subsets[].trafficPolicy.loadBalancer.localityLbSetting.failoverPriority[]

Type

`string`

.spec.subsets[].trafficPolicy.outlierDetection

Type

`object`

Property	Type	Description
`baseEjectionTime`	`string`	Minimum ejection duration.
`consecutive5xxErrors`	`integer`	Number of 5xx errors before a host is ejected from the connection pool.
`consecutiveErrors`	`integer`	
`consecutiveGatewayErrors`	`integer`	Number of gateway errors before a host is ejected from the connection pool.
`consecutiveLocalOriginFailures`	`integer`	
`interval`	`string`	Time interval between ejection sweep analysis.
`maxEjectionPercent`	`integer`	
`minHealthPercent`	`integer`	
`splitExternalLocalOriginErrors`	`boolean`	Determines whether to distinguish local origin failures from external errors.

.spec.subsets[].trafficPolicy.portLevelSettings

Description

Traffic policies specific to individual ports.

Type

`array`

.spec.subsets[].trafficPolicy.portLevelSettings[]

Type

`object`

Property	Type	Description
`connectionPool`	`object`	
`loadBalancer`	`object`	Settings controlling the load balancer algorithms.
`outlierDetection`	`object`	
`port`	`object`	
`tls`	`object`	TLS related settings for connections to the upstream service.

.spec.subsets[].trafficPolicy.portLevelSettings[].connectionPool

Type

`object`

Property	Type	Description
`http`	`object`	HTTP connection pool settings.

Property	Type	Description
`tcp`	`object`	Settings common to both HTTP and TCP upstream connections.

.spec.subsets[].trafficPolicy.portLevelSettings[].connectionPool.http

Description

HTTP connection pool settings.

Type

`object`

Property	Type	Description
`h2UpgradePolicy`	`string`	Specify if http1.1 connection should be upgraded to http2 for the associated destination.
`http1MaxPendingRequests`	`integer`	
`http2MaxRequests`	`integer`	Maximum number of active requests to a destination.
`idleTimeout`	`string`	The idle timeout for upstream connection pool connections.

Property	Type	Description
`maxRequestsPerConnection`	`integer`	Maximum number of requests per connection to a backend.
`maxRetries`	`integer`	
`useClientProtocol`	`boolean`	If set to true, client protocol will be preserved while initiating connection to backend.

.spec.subsets[].trafficPolicy.portLevelSettings[].connectionPool.tcp

Description

Settings common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
`connectTimeout`	`string`	TCP connection timeout.
`maxConnectionDuration`	`string`	The maximum duration of a connection.
`maxConnections`	`integer`	Maximum number of HTTP1 /TCP connections to a destination host.

Property	Type	Description
`tcpKeepalive`	`object`	If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

.spec.subsets[].trafficPolicy.portLevelSettings[].connectionPool.tcp.tcpKeepalive

Description

If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

Type

`object`

Property	Type	Description
`interval`	`string`	The time duration between keep-alive probes.
`probes`	`integer`	
`time`	`string`	

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer

Description

Settings controlling the load balancer algorithms.

Type

`object`

Property	Type	Description
`consistentHash`	`object`	
`localityLbSetting`	`object`	
`simple`	`string`	
`warmupDurationSecs`	`string`	Represents the warmup duration of Service.

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.consistentHash

Type

`object`

Property	Type	Description
`httpCookie`	`object`	Hash based on HTTP cookie.
`httpHeaderName`	`string`	Hash based on a specific HTTP header.
`httpQueryParameterName`	`string`	Hash based on a specific HTTP query parameter.
`maglev`	`object`	The Maglev load balancer implements consistent hashing to backend hosts.

Property	Type	Description
`minimumRingSize`	`integer`	Deprecated.
`ringHash`	`object`	The ring/modulo hash load balancer implements consistent hashing to backend hosts.
`useSourceIp`	`boolean`	Hash based on the source IP address.

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.httpCookie

Description

Hash based on HTTP cookie.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the cookie.
`path`	`string`	Path to set for the cookie.
`ttl`	`string`	Lifetime of the cookie.

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.maglev

Description

The Maglev load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`tableSize`	`integer`	The table size for Maglev hashing.

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.ringHash

Description

The ring/modulo hash load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`minimumRingSize`	`integer`	

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting

Type

`object`

Property	Type	Description
`distribute`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`enabled`	`boolean`	enable locality load balancing, this is DestinationRule-level and will override mesh wide settings in entirety.
`failover`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`failoverPriority`	`array`	failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.distribute

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.distribute[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating locality, '/' separated, e.g.
`to`	`object`	Map of upstream localities to traffic distribution weights.

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.distribute[].to

Description

Map of upstream localities to traffic distribution weights.

Type

`object`

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failover

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failover[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating region.
`to`	`string`	

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failoverPriority

Description

failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

Type

`array`

.spec.subsets[].trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failoverPriority[]

Type

`string`

.spec.subsets[].trafficPolicy.portLevelSettings[].outlierDetection

Type

`object`

Property	Type	Description
`baseEjectionTime`	`string`	Minimum ejection duration.
`consecutive5xxErrors`	`integer`	Number of 5xx errors before a host is ejected from the connection pool.
`consecutiveErrors`	`integer`	
`consecutiveGatewayErrors`	`integer`	Number of gateway errors before a host is ejected from the connection pool.
`consecutiveLocalOriginFailures`	`integer`	
`interval`	`string`	Time interval between ejection sweep analysis.
`maxEjectionPercent`	`integer`	
`minHealthPercent`	`integer`	
`splitExternalLocalOriginErrors`	`boolean`	Determines whether to distinguish local origin failures from external errors.

.spec.subsets[].trafficPolicy.portLevelSettings[].port

Type

`object`

Property	Type	Description
`number`	`integer`	

.spec.subsets[].trafficPolicy.portLevelSettings[].tls

Description

TLS related settings for connections to the upstream service.

Type

`object`

Property	Type	Description
`caCertificates`	`string`	
`clientCertificate`	`string`	REQUIRED if mode is `MUTUAL`.
`credentialName`	`string`	
`insecureSkipVerify`	`boolean`	
`mode`	`string`	
`privateKey`	`string`	REQUIRED if mode is `MUTUAL`.
`sni`	`string`	SNI string to present to the server during TLS handshake.
`subjectAltNames`	`array`	

.spec.subsets[].trafficPolicy.portLevelSettings[].tls.subjectAltNames

Type

`array`

.spec.subsets[].trafficPolicy.portLevelSettings[].tls.subjectAltNames[]

Type

`string`

.spec.subsets[].trafficPolicy.tls

Description

TLS related settings for connections to the upstream service.

Type

`object`

Property	Type	Description
`caCertificates`	`string`	
`clientCertificate`	`string`	REQUIRED if mode is `MUTUAL`.
`credentialName`	`string`	
`insecureSkipVerify`	`boolean`	
`mode`	`string`	

Property	Type	Description
`privateKey`	`string`	REQUIRED if mode is `MUTUAL`.
`sni`	`string`	SNI string to present to the server during TLS handshake.
`subjectAltNames`	`array`	

.spec.subsets[].trafficPolicy.tls.subjectAltNames

Type

`array`

.spec.subsets[].trafficPolicy.tls.subjectAltNames[]

Type

`string`

.spec.subsets[].trafficPolicy.tunnel

Type

`object`

Property	Type	Description
`protocol`	`string`	Specifies which protocol to use for tunneling the downstream connection.

Property	Type	Description
`targetHost`	`string`	Specifies a host to which the downstream connection is tunneled.
`targetPort`	`integer`	Specifies a port to which the downstream connection is tunneled.

.spec.trafficPolicy

Type

`object`

Property	Type	Description
`connectionPool`	`object`	
`loadBalancer`	`object`	Settings controlling the load balancer algorithms.
`outlierDetection`	`object`	
`portLevelSettings`	`array`	Traffic policies specific to individual ports.
`tls`	`object`	TLS related settings for connections to the upstream service.
`tunnel`	`object`	

.spec.trafficPolicy.connectionPool

Type

`object`

Property	Type	Description
`http`	`object`	HTTP connection pool settings.
`tcp`	`object`	Settings common to both HTTP and TCP upstream connections.

.spec.trafficPolicy.connectionPool.http

Description

HTTP connection pool settings.

Type

`object`

Property	Type	Description
`h2UpgradePolicy`	`string`	Specify if http1.1 connection should be upgraded to http2 for the associated destination.
`http1MaxPendingRequests`	`integer`	
`http2MaxRequests`	`integer`	Maximum number of active requests to a destination.

Property	Type	Description
`idleTimeout`	`string`	The idle timeout for upstream connection pool connections.
`maxRequestsPerConnection`	`integer`	Maximum number of requests per connection to a backend.
`maxRetries`	`integer`	
`useClientProtocol`	`boolean`	If set to true, client protocol will be preserved while initiating connection to backend.

.spec.trafficPolicy.connectionPool.tcp

Description

Settings common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
`connectTimeout`	`string`	TCP connection timeout.
`maxConnectionDuration`	`string`	The maximum duration of a connection.

Property	Type	Description
`maxConnections`	`integer`	Maximum number of HTTP1 /TCP connections to a destination host.
`tcpKeepalive`	`object`	If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

.spec.trafficPolicy.connectionPool.tcp.tcpKeepalive

Description

If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

Type

`object`

Property	Type	Description
`interval`	`string`	The time duration between keep-alive probes.
`probes`	`integer`	
`time`	`string`	

.spec.trafficPolicy.loadBalancer

Description

Settings controlling the load balancer algorithms.

Type

`object`

Property	Type	Description
`consistentHash`	`object`	
`localityLbSetting`	`object`	
`simple`	`string`	
`warmupDurationSecs`	`string`	Represents the warmup duration of Service.

.spec.trafficPolicy.loadBalancer.consistentHash

Type

`object`

Property	Type	Description
`httpCookie`	`object`	Hash based on HTTP cookie.
`httpHeaderName`	`string`	Hash based on a specific HTTP header.
`httpQueryParameterName`	`string`	Hash based on a specific HTTP query parameter.
`maglev`	`object`	The Maglev load balancer implements consistent hashing to backend hosts.
`minimumRingSize`	`integer`	Deprecated.

Property	Type	Description
`ringHash`	`object`	The ring/modulo hash load balancer implements consistent hashing to backend hosts.
`useSourceIp`	`boolean`	Hash based on the source IP address.

.spec.trafficPolicy.loadBalancer.consistentHash.httpCookie

Description

Hash based on HTTP cookie.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the cookie.
`path`	`string`	Path to set for the cookie.
`ttl`	`string`	Lifetime of the cookie.

.spec.trafficPolicy.loadBalancer.consistentHash.maglev

Description

The Maglev load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`tableSize`	`integer`	The table size for Maglev hashing.

.spec.trafficPolicy.loadBalancer.consistentHash.ringHash

Description

The ring/modulo hash load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`minimumRingSize`	`integer`	

.spec.trafficPolicy.loadBalancer.localityLbSetting

Type

`object`

Property	Type	Description
`distribute`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`enabled`	`boolean`	enable locality load balancing, this is DestinationRule-level and will override mesh wide

Property	Type	Description
		settings in entirety.
`failover`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`failoverPriority`	`array`	failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

.spec.trafficPolicy.loadBalancer.localityLbSetting.distribute

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.trafficPolicy.loadBalancer.localityLbSetting.distribute[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating locality, '/' separated, e.g.

Property	Type	Description
`to`	`object`	Map of upstream localities to traffic distribution weights.

.spec.trafficPolicy.loadBalancer.localityLbSetting.distribute[].to

Description

Map of upstream localities to traffic distribution weights.

Type

`object`

.spec.trafficPolicy.loadBalancer.localityLbSetting.failover

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.trafficPolicy.loadBalancer.localityLbSetting.failover[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating region.

Property	Type	Description
`to`	`string`	

.spec.trafficPolicy.loadBalancer.localityLbSetting.failoverPriority

Description

failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

Type

`array`

.spec.trafficPolicy.loadBalancer.localityLbSetting.failoverPriority[]

Type

`string`

.spec.trafficPolicy.outlierDetection

Type

`object`

Property	Type	Description
`baseEjectionTime`	`string`	Minimum ejection duration.
`consecutive5xxErrors`	`integer`	Number of 5xx errors before a host is ejected from the connection pool.

Property	Type	Description
`consecutiveErrors`	`integer`	
`consecutiveGatewayErrors`	`integer`	Number of gateway errors before a host is ejected from the connection pool.
`consecutiveLocalOriginFailures`	`integer`	
`interval`	`string`	Time interval between ejection sweep analysis.
`maxEjectionPercent`	`integer`	
`minHealthPercent`	`integer`	
`splitExternalLocalOriginErrors`	`boolean`	Determines whether to distinguish local origin failures from external errors.

.spec.trafficPolicy.portLevelSettings

Description

Traffic policies specific to individual ports.

Type

`array`

.spec.trafficPolicy.portLevelSettings[]

Type

`object`

Property	Type	Description
`connectionPool`	`object`	
`loadBalancer`	`object`	Settings controlling the load balancer algorithms.
`outlierDetection`	`object`	
`port`	`object`	
`tls`	`object`	TLS related settings for connections to the upstream service.

.spec.trafficPolicy.portLevelSettings[].connectionPool

Type

`object`

Property	Type	Description
`http`	`object`	HTTP connection pool settings.
`tcp`	`object`	Settings common to both HTTP and TCP upstream connections.

.spec.trafficPolicy.portLevelSettings[].connectionPool.http

Description

HTTP connection pool settings.

Type

`object`

Property	Type	Description
`h2UpgradePolicy`	`string`	Specify if http1.1 connection should be upgraded to http2 for the associated destination.
`http1MaxPendingRequests`	`integer`	
`http2MaxRequests`	`integer`	Maximum number of active requests to a destination.
`idleTimeout`	`string`	The idle timeout for upstream connection pool connections.
`maxRequestsPerConnection`	`integer`	Maximum number of requests per connection to a backend.
`maxRetries`	`integer`	
`useClientProtocol`	`boolean`	If set to true, client protocol will be preserved while initiating connection to backend.

.spec.trafficPolicy.portLevelSettings[].connectionPool.tcp

Description

Settings common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
`connectTimeout`	`string`	TCP connection timeout.
`maxConnectionDuration`	`string`	The maximum duration of a connection.
`maxConnections`	`integer`	Maximum number of HTTP1 /TCP connections to a destination host.
`tcpKeepalive`	`object`	If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

.spec.trafficPolicy.portLevelSettings[].connectionPool.tcp.tcpKeepalive

Description

If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

Type

`object`

Property	Type	Description
`interval`	`string`	The time duration between keep-alive probes.
`probes`	`integer`	
`time`	`string`	

.spec.trafficPolicy.portLevelSettings[].loadBalancer

Description

Settings controlling the load balancer algorithms.

Type

`object`

Property	Type	Description
`consistentHash`	`object`	
`localityLbSetting`	`object`	
`simple`	`string`	
`warmupDurationSecs`	`string`	Represents the warmup duration of Service.

.spec.trafficPolicy.portLevelSettings[].loadBalancer.consistentHash

Type

`object`

Property	Type	Description
`httpCookie`	`object`	Hash based on HTTP cookie.
`httpHeaderName`	`string`	Hash based on a specific HTTP header.
`httpQueryParameterName`	`string`	Hash based on a specific HTTP query parameter.
`maglev`	`object`	The Maglev load balancer implements consistent hashing to backend hosts.
`minimumRingSize`	`integer`	Deprecated.
`ringHash`	`object`	The ring/modulo hash load balancer implements consistent hashing to backend hosts.
`useSourceIp`	`boolean`	Hash based on the source IP address.

.spec.trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.httpCookie

Description

Hash based on HTTP cookie.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the cookie.
`path`	`string`	Path to set for the cookie.
`ttl`	`string`	Lifetime of the cookie.

.spec.trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.maglev

Description

The Maglev load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`tableSize`	`integer`	The table size for Maglev hashing.

.spec.trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.ringHash

Description

The ring/modulo hash load balancer implements consistent hashing to backend hosts.

Type

`object`

Property	Type	Description
`minimumRingSize`	`integer`	

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting

Type

`object`

Property	Type	Description
`distribute`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`enabled`	`boolean`	enable locality load balancing, this is DestinationRule-level and will override mesh wide settings in entirety.
`failover`	`array`	Optional: only one of distribute, failover or failoverPriority can be set.
`failoverPriority`	`array`	failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.distribute

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.distribute[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating locality, '/' separated, e.g.
`to`	`object`	Map of upstream localities to traffic distribution weights.

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.distribute[].to

Description

Map of upstream localities to traffic distribution weights.

Type

`object`

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failover

Description

Optional: only one of distribute, failover or failoverPriority can be set.

Type

`array`

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failover[]

Type

`object`

Property	Type	Description
`from`	`string`	Originating region.
`to`	`string`	

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failoverPriority

Description

failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing.

Type

`array`

.spec.trafficPolicy.portLevelSettings[].loadBalancer.localityLbSetting.failoverPriority[]

Type

`string`

.spec.trafficPolicy.portLevelSettings[].outlierDetection

Type

`object`

Property	Type	Description
`baseEjectionTime`	`string`	Minimum ejection duration.
`consecutive5xxErrors`	`integer`	Number of 5xx errors before a host is ejected from the connection pool.
`consecutiveErrors`	`integer`	
`consecutiveGatewayErrors`	`integer`	Number of gateway errors before a host is ejected from the connection pool.
`consecutiveLocalOriginFailures`	`integer`	
`interval`	`string`	Time interval between ejection sweep analysis.
`maxEjectionPercent`	`integer`	
`minHealthPercent`	`integer`	
`splitExternalLocalOriginErrors`	`boolean`	Determines whether to distinguish local origin failures from external errors.

.spec.trafficPolicy.portLevelSettings[].port

Type

`object`

Property	Type	Description
`number`	`integer`	

.spec.trafficPolicy.portLevelSettings[].tls

Description

TLS related settings for connections to the upstream service.

Type

`object`

Property	Type	Description
`caCertificates`	`string`	
`clientCertificate`	`string`	REQUIRED if mode is `MUTUAL`.
`credentialName`	`string`	
`insecureSkipVerify`	`boolean`	
`mode`	`string`	
`privateKey`	`string`	REQUIRED if mode is `MUTUAL`.

Property	Type	Description
`sni`	`string`	SNI string to present to the server during TLS handshake.
`subjectAltNames`	`array`	

.spec.trafficPolicy.portLevelSettings[].tls.subjectAltNames

Type

`array`

.spec.trafficPolicy.portLevelSettings[].tls.subjectAltNames[]

Type

`string`

.spec.trafficPolicy.tls

Description

TLS related settings for connections to the upstream service.

Type

`object`

Property	Type	Description
`caCertificates`	`string`	
`clientCertificate`	`string`	REQUIRED if mode is `MUTUAL`.

Property	Type	Description
`credentialName`	`string`	
`insecureSkipVerify`	`boolean`	
`mode`	`string`	
`privateKey`	`string`	REQUIRED if mode is `MUTUAL`.
`sni`	`string`	SNI string to present to the server during TLS handshake.
`subjectAltNames`	`array`	

.spec.trafficPolicy.tls.subjectAltNames

Type

`array`

.spec.trafficPolicy.tls.subjectAltNames[]

Type

`string`

.spec.trafficPolicy.tunnel

Type

`object`

Property	Type	Description
`protocol`	`string`	Specifies which protocol to use for tunneling the downstream connection.
`targetHost`	`string`	Specifies a host to which the downstream connection is tunneled.
`targetPort`	`integer`	Specifies a port to which the downstream connection is tunneled.

.spec.workloadSelector

Type

`object`

Property	Type	Description
`matchLabels`	`object`	

.spec.workloadSelector.matchLabels

Type

`object`

.status

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/destinationrules`
  - `DELETE`: delete collection of DestinationRule
  - `GET`: list objects of kind DestinationRule
  - `POST`: create a new DestinationRule
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/destinationrules/{name}`
  - `DELETE`: delete the specified DestinationRule
  - `GET`: read the specified DestinationRule
  - `PATCH`: partially update the specified DestinationRule
  - `PUT`: replace the specified DestinationRule
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/destinationrules/{name}/status`
  - `GET`: read status of the specified DestinationRule
  - `PATCH`: partially update status of the specified DestinationRule
  - `PUT`: replace status of the specified DestinationRule

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/destinationrules

HTTP method

`DELETE`

Description

delete collection of DestinationRule

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind DestinationRule

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRuleList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new DestinationRule

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`DestinationRule` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
201 - Created	`DestinationRule` schema
202 - Accepted	`DestinationRule` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/destinationrules/{name}

HTTP method

`DELETE`

Description

delete the specified DestinationRule

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	`Status↗` schema
202 - Accepted	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified DestinationRule

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified DestinationRule

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified DestinationRule

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`DestinationRule` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
201 - Created	`DestinationRule` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/destinationrules/{name}/status

HTTP method

`GET`

Description

read status of the specified DestinationRule

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified DestinationRule

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified DestinationRule

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`DestinationRule` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`DestinationRule` schema
201 - Created	`DestinationRule` schema

HTTP code	Response body
401 - Unauthorized	Empty

Type

`object`

Specification

Property	Type	Description
`spec`	`object`	Configuration affecting edge load balancer. See more details at: https://istio.io/docs/reference/config/networking/gateway.html ↗
`status`	`object`	

.spec

Description

Configuration affecting edge load balancer. See more details at: https://istio.io/docs/reference/config/networking/gateway.html

Type

`object`

Property	Type	Description
`selector`	`object`	
`servers`	`array`	A list of server specifications.

.spec.selector

Type

`object`

.spec.servers

Description

A list of server specifications.

Type

`array`

.spec.servers[]

Type

`object`

Property	Type	Description
`bind`	`string`	
`defaultEndpoint`	`string`	
`hosts`	`array`	One or more hosts exposed by this gateway.

Property	Type	Description
`name`	`string`	An optional name of the server, when set must be unique across all servers.
`port`	`object`	
`tls`	`object`	Set of TLS related options that govern the server's behavior.

.spec.servers[].hosts

Description

One or more hosts exposed by this gateway.

Type

`array`

.spec.servers[].hosts[]

Type

`string`

.spec.servers[].port

Type

`object`

Property	Type	Description
`name`	`string`	Label assigned to the port.
`number`	`integer`	A valid non-negative integer port number.
`protocol`	`string`	The protocol exposed on the port.
`targetPort`	`integer`	

.spec.servers[].tls

Description

Set of TLS related options that govern the server's behavior.

Type

`object`

Property	Type	Description
`caCertificates`	`string`	REQUIRED if mode is `MUTUAL`.
`cipherSuites`	`array`	Optional: If specified, only support the specified cipher list.
`credentialName`	`string`	
`httpsRedirect`	`boolean`	

Property	Type	Description
`maxProtocolVersion`	`string`	Optional: Maximum TLS protocol version.
`minProtocolVersion`	`string`	Optional: Minimum TLS protocol version.
`mode`	`string`	
`privateKey`	`string`	REQUIRED if mode is `SIMPLE` or `MUTUAL`.
`serverCertificate`	`string`	REQUIRED if mode is `SIMPLE` or `MUTUAL`.
`subjectAltNames`	`array`	
`verifyCertificateHash`	`array`	
`verifyCertificateSpki`	`array`	

.spec.servers[].tls.cipherSuites

Description

Optional: If specified, only support the specified cipher list.

Type

`array`

.spec.servers[].tls.cipherSuites[]

Type

`string`

.spec.servers[].tls.subjectAltNames

Type

`array`

.spec.servers[].tls.subjectAltNames[]

Type

`string`

.spec.servers[].tls.verifyCertificateHash

Type

`array`

.spec.servers[].tls.verifyCertificateHash[]

Type

`string`

.spec.servers[].tls.verifyCertificateSpki

Type

`array`

.spec.servers[].tls.verifyCertificateSpki[]

Type

`string`

.status

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/gateways`
  - `DELETE`: delete collection of Gateway
  - `GET`: list objects of kind Gateway
  - `POST`: create a new Gateway
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/gateways/{name}`
  - `DELETE`: delete the specified Gateway
  - `GET`: read the specified Gateway
  - `PATCH`: partially update the specified Gateway
  - `PUT`: replace the specified Gateway
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/gateways/{name}/status`
  - `GET`: read status of the specified Gateway
  - `PATCH`: partially update status of the specified Gateway
  - `PUT`: replace status of the specified Gateway

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/gateways

HTTP method

`DELETE`

Description

delete collection of Gateway

HTTP responses

HTTP code	Response body
200 - OK	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Gateway

HTTP responses

HTTP code	Response body
200 - OK	`GatewayList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Gateway

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Gateway` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
201 - Created	`Gateway` schema
202 - Accepted	`Gateway` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/gateways/{name}

HTTP method

`DELETE`

Description

delete the specified Gateway

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	`Status↗` schema
202 - Accepted	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified Gateway

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Gateway

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Gateway

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Gateway` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
201 - Created	`Gateway` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/gateways/{name}/status

HTTP method

`GET`

Description

read status of the specified Gateway

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Gateway

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Gateway

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Gateway` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Gateway` schema
201 - Created	`Gateway` schema

HTTP code	Response body
401 - Unauthorized	Empty

Type

`object`

Specification

Property	Type	Description
`spec`	`object`	Configuration affecting network reachability of a sidecar. See more details at: https://istio.io/docs/reference/config/networking/sidecar.html ↗
`status`	`object`	

.spec

Description

Configuration affecting network reachability of a sidecar. See more details at: https://istio.io/docs/reference/config/networking/sidecar.html

Type

`object`

Property	Type	Description
`egress`	`array`	
`ingress`	`array`	
`outboundTrafficPolicy`	`object`	Configuration for the outbound traffic policy.
`workloadSelector`	`object`	

.spec.egress

Type

`array`

.spec.egress[]

Type

`object`

Property	Type	Description
`bind`	`string`	
`captureMode`	`string`	
`hosts`	`array`	
`port`	`object`	The port associated with the listener.

.spec.egress[].hosts

Type

`array`

.spec.egress[].hosts[]

Type

`string`

.spec.egress[].port

Description

The port associated with the listener.

Type

`object`

Property	Type	Description
`name`	`string`	Label assigned to the port.
`number`	`integer`	A valid non-negative integer port number.
`protocol`	`string`	The protocol exposed on the port.
`targetPort`	`integer`	

.spec.ingress

Type

`array`

.spec.ingress[]

Type

`object`

Property	Type	Description
`bind`	`string`	The IP(IPv4 or IPv6) to which the listener should be bound.
`captureMode`	`string`	
`defaultEndpoint`	`string`	
`port`	`object`	The port associated with the listener.
`tls`	`object`	

.spec.ingress[].port

Description

The port associated with the listener.

Type

`object`

Property	Type	Description
`name`	`string`	Label assigned to the port.
`number`	`integer`	A valid non-negative integer port number.

Property	Type	Description
`protocol`	`string`	The protocol exposed on the port.
`targetPort`	`integer`	

.spec.ingress[].tls

Type

`object`

Property	Type	Description
`caCertificates`	`string`	REQUIRED if mode is `MUTUAL`.
`cipherSuites`	`array`	Optional: If specified, only support the specified cipher list.
`credentialName`	`string`	
`httpsRedirect`	`boolean`	
`maxProtocolVersion`	`string`	Optional: Maximum TLS protocol version.
`minProtocolVersion`	`string`	Optional: Minimum TLS protocol version.
`mode`	`string`	

Property	Type	Description
`privateKey`	`string`	REQUIRED if mode is `SIMPLE` or `MUTUAL`.
`serverCertificate`	`string`	REQUIRED if mode is `SIMPLE` or `MUTUAL`.
`subjectAltNames`	`array`	
`verifyCertificateHash`	`array`	
`verifyCertificateSpki`	`array`	

.spec.ingress[].tls.cipherSuites

Description

Optional: If specified, only support the specified cipher list.

Type

`array`

.spec.ingress[].tls.cipherSuites[]

Type

`string`

.spec.ingress[].tls.subjectAltNames

Type

`array`

.spec.ingress[].tls.subjectAltNames[]

Type

`string`

.spec.ingress[].tls.verifyCertificateHash

Type

`array`

.spec.ingress[].tls.verifyCertificateHash[]

Type

`string`

.spec.ingress[].tls.verifyCertificateSpki

Type

`array`

.spec.ingress[].tls.verifyCertificateSpki[]

Type

`string`

.spec.outboundTrafficPolicy

Description

Configuration for the outbound traffic policy.

Type

`object`

Property	Type	Description
`egressProxy`	`object`	
`mode`	`string`	

.spec.outboundTrafficPolicy.egressProxy

Type

`object`

Property	Type	Description
`host`	`string`	The name of a service from the service registry.
`port`	`object`	Specifies the port on the host that is being addressed.
`subset`	`string`	The name of a subset within the service.

.spec.outboundTrafficPolicy.egressProxy.port

Description

Specifies the port on the host that is being addressed.

Type

`object`

Property	Type	Description
`number`	`integer`	

.spec.workloadSelector

Type

`object`

Property	Type	Description
`labels`	`object`	

.spec.workloadSelector.labels

Type

`object`

.status

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/sidecars`
  - `DELETE`: delete collection of Sidecar
  - `GET`: list objects of kind Sidecar
  - `POST`: create a new Sidecar
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/sidecars/{name}`
  - `DELETE`: delete the specified Sidecar
  - `GET`: read the specified Sidecar
  - `PATCH`: partially update the specified Sidecar

- `PUT`: replace the specified Sidecar

- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/sidecars/{name}/status`
  - `GET`: read status of the specified Sidecar
  - `PATCH`: partially update status of the specified Sidecar
  - `PUT`: replace status of the specified Sidecar

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/sidecars

HTTP method

`DELETE`

Description

delete collection of Sidecar

HTTP responses

HTTP code	Response body
200 - OK	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Sidecar

HTTP responses

HTTP code	Response body
200 - OK	`SidecarList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Sidecar

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Sidecar` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
201 - Created	`Sidecar` schema
202 - Accepted	`Sidecar` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/sidecars/{name}

HTTP method

`DELETE`

Description

delete the specified Sidecar

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
202 - Accepted	`Status` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified Sidecar

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Sidecar

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

Parameter	Type	Description
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Sidecar

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently

Parameter	Type	Description
		dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Sidecar` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
201 - Created	`Sidecar` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/sidecars/{name}/status

HTTP method

`GET`

Description

read status of the specified Sidecar

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Sidecar

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Sidecar

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Sidecar` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Sidecar` schema
201 - Created	`Sidecar` schema
401 - Unauthorized	Empty

Type

`object`

Specification

Property	Type	Description
`spec`	`object`	Configuration affecting label/content routing, sni routing, etc. See more details at: https://istio.io/docs/reference/config/networking/virtual-service.html ↗
`status`	`object`	

.spec

Description

Configuration affecting label/content routing, sni routing, etc. See more details at: https://istio.io/docs/reference/config/networking/virtual-service.html

Type

`object`

Property	Type	Description
`exportTo`	`array`	A list of namespaces to which this virtual service is exported.
`gateways`	`array`	The names of gateways and sidecars that should apply these routes.
`hosts`	`array`	The destination hosts to which traffic is being sent.
`http`	`array`	An ordered list of route rules for HTTP traffic.
`tcp`	`array`	An ordered list of route rules for opaque TCP traffic.
`tls`	`array`	

.spec.exportTo

Description

A list of namespaces to which this virtual service is exported.

Type

`array`

.spec.exportTo[]

Type

`string`

.spec.gateways

Description

The names of gateways and sidecars that should apply these routes.

Type

`array`

.spec.gateways[]

Type

`string`

.spec.hosts

Description

The destination hosts to which traffic is being sent.

Type

`array`

.spec.hosts[]

Type

`string`

.spec.http

Description

An ordered list of route rules for HTTP traffic.

Type

`array`

.spec.http[]

Type

`object`

Property	Type	Description
`corsPolicy`	`object`	Cross-Origin Resource Sharing policy (CORS).
`delegate`	`object`	
`directResponse`	`object`	A HTTP rule can either return a direct_response, redirect or forward (default) traffic.
`fault`	`object`	Fault injection policy to apply on HTTP traffic at the client side.
`headers`	`object`	
`match`	`array`	
`mirror`	`object`	
`mirror_percent`	`integer`	Percentage of the traffic to be mirrored by the `mirror` field.
`mirrorPercent`	`integer`	Percentage of the traffic to be mirrored by the `mirror` field.

Property	Type	Description
`mirrorPercentage`	`object`	Percentage of the traffic to be mirrored by the `mirror` field.
`name`	`string`	The name assigned to the route for debugging purposes.
`redirect`	`object`	A HTTP rule can either return a direct_response, redirect or forward (default) traffic.
`retries`	`object`	Retry policy for HTTP requests.
`rewrite`	`object`	Rewrite HTTP URIs and Authority headers.
`route`	`array`	A HTTP rule can either return a direct_response, redirect or forward (default) traffic.
`timeout`	`string`	Timeout for HTTP requests, default is disabled.

.spec.http[].corsPolicy

Description

Cross-Origin Resource Sharing policy (CORS).

Type

`object`

Property	Type	Description
`allowCredentials`	`boolean`	
`allowHeaders`	`array`	
`allowMethods`	`array`	List of HTTP methods allowed to access the resource.
`allowOrigin`	`array`	The list of origins that are allowed to perform CORS requests.
`allowOrigins`	`array`	String patterns that match allowed origins.
`exposeHeaders`	`array`	
`maxAge`	`string`	

.spec.http[].corsPolicy.allowHeaders

Type

`array`

.spec.http[].corsPolicy.allowHeaders[]

Type

`string`

.spec.http[].corsPolicy.allowMethods

Description

List of HTTP methods allowed to access the resource.

Type

`array`

.spec.http[].corsPolicy.allowMethods[]

Type

`string`

.spec.http[].corsPolicy.allowOrigin

Description

The list of origins that are allowed to perform CORS requests.

Type

`array`

.spec.http[].corsPolicy.allowOrigin[]

Type

`string`

.spec.http[].corsPolicy.allowOrigins

Description

String patterns that match allowed origins.

Type

`array`

.spec.http[].corsPolicy.allowOrigins[]

Type

`object`

Property	Type	Description
`exact`	`string`	
`prefix`	`string`	
`regex`	`string`	RE2 style regex-based match (https://github.com/google/re2/wiki/Syntax↗).

.spec.http[].corsPolicy.exposeHeaders

Type

`array`

.spec.http[].corsPolicy.exposeHeaders[]

Type

`string`

.spec.http[].delegate

Type

`object`

Property	Type	Description
`name`	`string`	Name specifies the name of the delegate VirtualService.
`namespace`	`string`	Namespace specifies the namespace where the delegate VirtualService resides.

.spec.http[].directResponse

Description

A HTTP rule can either return a direct_response, redirect or forward (default) traffic.

Type

`object`

Property	Type	Description
`body`	`object`	Specifies the content of the response body.
`status`	`integer`	Specifies the HTTP response status to be returned.

.spec.http[].directResponse.body

Description

Specifies the content of the response body.

Type

`object`

Property	Type	Description
`bytes`	`string`	response body as base64 encoded bytes.
`string`	`string`	

.spec.http[].fault

Description

Fault injection policy to apply on HTTP traffic at the client side.

Type

`object`

Property	Type	Description
`abort`	`object`	
`delay`	`object`	

.spec.http[].fault.abort

Type

`object`

Property	Type	Description
`grpcStatus`	`string`	GRPC status code to use to abort the request.
`http2Error`	`string`	
`httpStatus`	`integer`	HTTP status code to use to abort the Http request.
`percentage`	`object`	Percentage of requests to be aborted with the error code provided.

.spec.http[].fault.abort.percentage

Description

Percentage of requests to be aborted with the error code provided.

Type

`object`

Property	Type	Description
`value`	`number`	

.spec.http[].fault.delay

Type

`object`

Property	Type	Description
`exponentialDelay`	`string`	
`fixedDelay`	`string`	Add a fixed delay before forwarding the request.
`percent`	`integer`	Percentage of requests on which the delay will be injected (0-100).
`percentage`	`object`	Percentage of requests on which the delay will be injected.

.spec.http[].fault.delay.percentage

Description

Percentage of requests on which the delay will be injected.

Type

`object`

Property	Type	Description
`value`	`number`	

.spec.http[].headers

Type

`object`

Property	Type	Description
`request`	`object`	
`response`	`object`	

.spec.http[].headers.request

Type

`object`

Property	Type	Description
`add`	`object`	
`remove`	`array`	
`set`	`object`	

.spec.http[].headers.request.add

Type

`object`

.spec.http[].headers.request.remove

Type

`array`

.spec.http[].headers.request.remove[]

Type

`string`

.spec.http[].headers.request.set

Type

`object`

.spec.http[].headers.response

Type

`object`

Property	Type	Description
`add`	`object`	
`remove`	`array`	
`set`	`object`	

.spec.http[].headers.response.add

Type

`object`

.spec.http[].headers.response.remove

Type

`array`

.spec.http[].headers.response.remove[]

Type

`string`

.spec.http[].headers.response.set

Type

`object`

.spec.http[].match

Type

`array`

.spec.http[].match[]

Type

`object`

Property	Type	Description
`authority`	`object`	
`gateways`	`array`	Names of gateways where the rule should be applied.
`headers`	`object`	

Property	Type	Description
`ignoreUriCase`	`boolean`	Flag to specify whether the URI matching should be case-insensitive.
`method`	`object`	
`name`	`string`	The name assigned to a match.
`port`	`integer`	Specifies the ports on the host that is being addressed.
`queryParams`	`object`	Query parameters for matching.
`scheme`	`object`	
`sourceLabels`	`object`	
`sourceNamespace`	`string`	Source namespace constraining the applicability of a rule to workloads in that namespace.
`statPrefix`	`string`	The human readable prefix to use when emitting statistics for this route.
`uri`	`object`	
`withoutHeaders`	`object`	withoutHeader has the same syntax with the header, but has opposite meaning.

.spec.http[].match[].authority

Type

`object`

Property	Type	Description
`exact`	`string`	
`prefix`	`string`	
`regex`	`string`	RE2 style regex-based match (https://github.com/google/re2/wiki/Syntax ↗).

.spec.http[].match[].gateways

Description

Names of gateways where the rule should be applied.

Type

`array`

.spec.http[].match[].gateways[]

Type

`string`

.spec.http[].match[].headers

Type

`object`

.spec.http[].match[].method

Type

`object`

Property	Type	Description
`exact`	`string`	
`prefix`	`string`	
`regex`	`string`	RE2 style regex-based match (https://github.com/google/re2/wiki/Syntax ↗).

.spec.http[].match[].queryParams

Description

Query parameters for matching.

Type

`object`

.spec.http[].match[].scheme

Type

`object`

Property	Type	Description
`exact`	`string`	
`prefix`	`string`	
`regex`	`string`	RE2 style regex-based match (https://github.com/google/re2/wiki/Syntax ↗).

.spec.http[].match[].sourceLabels

Type

`object`

.spec.http[].match[].uri

Type

`object`

Property	Type	Description
`exact`	`string`	
`prefix`	`string`	
`regex`	`string`	RE2 style regex-based match (https://github.com/google/re2/wiki/Syntax ↗).

.spec.http[].match[].withoutHeaders

Description

withoutHeader has the same syntax with the header, but has opposite meaning.

Type

`object`

.spec.http[].mirror

Type

`object`

Property	Type	Description
`host`	`string`	The name of a service from the service registry.
`port`	`object`	Specifies the port on the host that is being addressed.
`subset`	`string`	The name of a subset within the service.

.spec.http[].mirror.port

Description

Specifies the port on the host that is being addressed.

Type

`object`

Property	Type	Description
`number`	`integer`	

.spec.http[].mirrorPercentage

Description

Percentage of the traffic to be mirrored by the `mirror` field.

Type

`object`

Property	Type	Description
`value`	`number`	

.spec.http[].redirect

Description

A HTTP rule can either return a direct_response, redirect or forward (default) traffic.

Type

`object`

Property	Type	Description
`authority`	`string`	
`derivePort`	`string`	
`port`	`integer`	On a redirect, overwrite the port portion of the URL with this value.
`redirectCode`	`integer`	
`scheme`	`string`	On a redirect, overwrite the scheme portion of the URL with this value.
`uri`	`string`	

.spec.http[].retries

Description

Retry policy for HTTP requests.

Type

`object`

Property	Type	Description
`attempts`	`integer`	Number of retries to be allowed for a given request.
`perTryTimeout`	`string`	Timeout per attempt for a given request, including the initial call and any retries.
`retryOn`	`string`	Specifies the conditions under which retry takes place.
`retryRemoteLocalities`	`boolean`	Flag to specify whether the retries should retry to other localities.

.spec.http[].rewrite

Description

Rewrite HTTP URIs and Authority headers.

Type

`object`

Property	Type	Description
`authority`	`string`	rewrite the Authority/Host header with this value.
`uri`	`string`	

.spec.http[].route

Description

A HTTP rule can either return a direct_response, redirect or forward (default) traffic.

Type

`array`

.spec.http[].route[]

Type

`object`

Property	Type	Description
`destination`	`object`	
`headers`	`object`	
`weight`	`integer`	Weight specifies the relative proportion of traffic to be forwarded to the destination.

.spec.http[].route[].destination

Type

`object`

Property	Type	Description
`host`	`string`	The name of a service from the service registry.

Property	Type	Description
`port`	`object`	Specifies the port on the host that is being addressed.
`subset`	`string`	The name of a subset within the service.

.spec.http[].route[].destination.port

Description

Specifies the port on the host that is being addressed.

Type

`object`

Property	Type	Description
`number`	`integer`	

.spec.http[].route[].headers

Type

`object`

Property	Type	Description
`request`	`object`	
`response`	`object`	

.spec.http[].route[].headers.request

Type

`object`

Property	Type	Description
`add`	`object`	
`remove`	`array`	
`set`	`object`	

.spec.http[].route[].headers.request.add

Type

`object`

.spec.http[].route[].headers.request.remove

Type

`array`

.spec.http[].route[].headers.request.remove[]

Type

`string`

.spec.http[].route[].headers.request.set

Type

`object`

.spec.http[].route[].headers.response

Type

`object`

Property	Type	Description
`add`	`object`	
`remove`	`array`	
`set`	`object`	

.spec.http[].route[].headers.response.add

Type

`object`

.spec.http[].route[].headers.response.remove

Type

`array`

.spec.http[].route[].headers.response.remove[]

Type

`string`

.spec.http[].route[].headers.response.set

Type

`object`

.spec.tcp

Description

An ordered list of route rules for opaque TCP traffic.

Type

`array`

.spec.tcp[]

Type

`object`

Property	Type	Description
`match`	`array`	
`route`	`array`	The destination to which the connection should be forwarded to.

.spec.tcp[].match

Type

`array`

.spec.tcp[].match[]

Type

`object`

Property	Type	Description
`destinationSubnets`	`array`	IPv4 or IPv6 ip addresses of destination with optional subnet.

Property	Type	Description
`gateways`	`array`	Names of gateways where the rule should be applied.
`port`	`integer`	Specifies the port on the host that is being addressed.
`sourceLabels`	`object`	
`sourceNamespace`	`string`	Source namespace constraining the applicability of a rule to workloads in that namespace.
`sourceSubnet`	`string`	IPv4 or IPv6 ip address of source with optional subnet.

.spec.tcp[].match[].destinationSubnets

Description

IPv4 or IPv6 ip addresses of destination with optional subnet.

Type

`array`

.spec.tcp[].match[].destinationSubnets[]

Type

`string`

.spec.tcp[].match[].gateways

Description

Names of gateways where the rule should be applied.

Type

`array`

.spec.tcp[].match[].gateways[]

Type

`string`

.spec.tcp[].match[].sourceLabels

Type

`object`

.spec.tcp[].route

Description

The destination to which the connection should be forwarded to.

Type

`array`

.spec.tcp[].route[]

Type

`object`

Property	Type	Description
`destination`	`object`	

Property	Type	Description
`weight`	`integer`	Weight specifies the relative proportion of traffic to be forwarded to the destination.

.spec.tcp[].route[].destination

Type

`object`

Property	Type	Description
`host`	`string`	The name of a service from the service registry.
`port`	`object`	Specifies the port on the host that is being addressed.
`subset`	`string`	The name of a subset within the service.

.spec.tcp[].route[].destination.port

Description

Specifies the port on the host that is being addressed.

Type

`object`

Property	Type	Description
`number`	`integer`	

.spec.tls

Type

`array`

.spec.tls[]

Type

`object`

Property	Type	Description
`match`	`array`	
`route`	`array`	The destination to which the connection should be forwarded to.

.spec.tls[].match

Type

`array`

.spec.tls[].match[]

Type

`object`

Property	Type	Description
`destinationSubnets`	`array`	IPv4 or IPv6 ip addresses of destination with optional subnet.

Property	Type	Description
`gateways`	`array`	Names of gateways where the rule should be applied.
`port`	`integer`	Specifies the port on the host that is being addressed.
`sniHosts`	`array`	SNI (server name indicator) to match on.
`sourceLabels`	`object`	
`sourceNamespace`	`string`	Source namespace constraining the applicability of a rule to workloads in that namespace.

.spec.tls[].match[].destinationSubnets

Description

IPv4 or IPv6 ip addresses of destination with optional subnet.

Type

`array`

.spec.tls[].match[].destinationSubnets[]

Type

`string`

.spec.tls[].match[].gateways

Description

Names of gateways where the rule should be applied.

Type

`array`

.spec.tls[].match[].gateways[]

Type

`string`

.spec.tls[].match[].sniHosts

Description

SNI (server name indicator) to match on.

Type

`array`

.spec.tls[].match[].sniHosts[]

Type

`string`

.spec.tls[].match[].sourceLabels

Type

`object`

.spec.tls[].route

Description

The destination to which the connection should be forwarded to.

Type

`array`

.spec.tls[].route[]

Type

`object`

Property	Type	Description
`destination`	`object`	
`weight`	`integer`	Weight specifies the relative proportion of traffic to be forwarded to the destination.

.spec.tls[].route[].destination

Type

`object`

Property	Type	Description
`host`	`string`	The name of a service from the service registry.
`port`	`object`	Specifies the port on the host that is being addressed.
`subset`	`string`	The name of a subset within the service.

.spec.tls[].route[].destination.port

Description

Specifies the port on the host that is being addressed.

Type

`object`

Property	Type	Description
`number`	`integer`	

.status

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/virtualservices`
  - `DELETE`: delete collection of VirtualService
  - `GET`: list objects of kind VirtualService
  - `POST`: create a new VirtualService
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/virtualservices/{name}`
  - `DELETE`: delete the specified VirtualService
  - `GET`: read the specified VirtualService
  - `PATCH`: partially update the specified VirtualService
  - `PUT`: replace the specified VirtualService
- `/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/virtualservices/{name}/status`

- `GET`: read status of the specified VirtualService
- `PATCH`: partially update status of the specified VirtualService
- `PUT`: replace status of the specified VirtualService

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/virtualservices

HTTP method

`DELETE`

Description

delete collection of VirtualService

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind VirtualService

HTTP responses

HTTP code	Response body
200 - OK	`VirtualServiceList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new VirtualService

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`VirtualService` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
201 - Created	`VirtualService` schema
202 - Accepted	`VirtualService` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/virtualservices/{name}

HTTP method

`DELETE`

Description

delete the specified VirtualService

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified VirtualService

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified VirtualService

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified VirtualService

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`VirtualService` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
201 - Created	`VirtualService` schema
401 - Unauthorized	Empty

/apis/networking.istio.io/v1alpha3/namespaces/{namespace}/virtualservices/{name}/status

HTTP method

`GET`

Description

read status of the specified VirtualService

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified VirtualService

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified VirtualService

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`VirtualService` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`VirtualService` schema
201 - Created	`VirtualService` schema
401 - Unauthorized	Empty