#

# From keystonemiddleware.auth\_token

#

# Complete "public" Identity API endpoint. This endpoint should not be an

# "admin" endpoint, as it should be accessible by all end users.

# Unauthenticated clients are redirected to this endpoint to authenticate.

# Although this endpoint should ideally be unversioned, client support in the

# wild varies. If you're using a versioned v2 endpoint here, then this should

# \*not\* be the same endpoint the service user utilizes for validating tokens,

# because normal end users may not be able to reach that endpoint. (string

# value)

# Deprecated group/name - [keystone\_authtoken]/auth\_uri

#www\_authenticate\_uri = <None>

www\_authenticate\_uri = {{ \_data.get('protocol', 'http') }}://{{ \_data.host }}:5000

# DEPRECATED: Complete "public" Identity API endpoint. This endpoint should not

# be an "admin" endpoint, as it should be accessible by all end users.

# Unauthenticated clients are redirected to this endpoint to authenticate.

# Although this endpoint should ideally be unversioned, client support in the

# wild varies. If you're using a versioned v2 endpoint here, then this should

# \*not\* be the same endpoint the service user utilizes for validating tokens,

# because normal end users may not be able to reach that endpoint. This option

# is deprecated in favor of www\_authenticate\_uri and will be removed in the S

# release. (string value)

# This option is deprecated for removal since Queens.

# Its value may be silently ignored in the future.

# Reason: The auth\_uri option is deprecated in favor of www\_authenticate\_uri

# and will be removed in the S release.

#auth\_uri = <None>

auth\_uri = {{ \_data.get('protocol', 'http') }}://{{ \_data.host }}:5000

# API version of the admin Identity API endpoint. (string value)

#auth\_version = <None>

{%- if \_data.get('version') %}

auth\_version = {{ \_data.version }}

{%- endif %}

# Do not handle authorization requests within the middleware, but delegate the

# authorization decision to downstream WSGI components. (boolean value)

#delay\_auth\_decision = false

# Request timeout value for communicating with Identity API server. (integer

# value)

#http\_connect\_timeout = <None>

# How many times are we trying to reconnect when communicating with Identity

# API Server. (integer value)

#http\_request\_max\_retries = 3

# Request environment key where the Swift cache object is stored. When

# auth\_token middleware is deployed with a Swift cache, use this option to have

# the middleware share a caching backend with swift. Otherwise, use the

# ``memcached\_servers`` option instead. (string value)

#cache = <None>

# Required if identity server requires client certificate (string value)

#certfile = <None>

# Required if identity server requires client certificate (string value)

#keyfile = <None>

# A PEM encoded Certificate Authority to use when verifying HTTPs connections.

# Defaults to system CAs. (string value)

#cafile = <None>

{%- if \_data.get('protocol', 'http') == 'https' %}

cafile={{ \_data.cacert\_file }}

{%- endif %}

# Verify HTTPS connections. (boolean value)

#insecure = false

# The region in which the identity server can be found. (string value)

#region\_name = <None>

{%- if \_data.region is defined %}

region\_name = {{ \_data.region }}

{%- endif %}

# DEPRECATED: Directory used to cache files related to PKI tokens. This option

# has been deprecated in the Ocata release and will be removed in the P

# release. (string value)

# This option is deprecated for removal since Ocata.

# Its value may be silently ignored in the future.

# Reason: PKI token format is no longer supported.

#signing\_dir = <None>

# Optionally specify a list of memcached server(s) to use for caching. If left

# undefined, tokens will instead be cached in-process. (list value)

# Deprecated group/name - [keystone\_authtoken]/memcache\_servers

#memcached\_servers = <None>

{%- if \_data.get('cache', {}).members is defined %}

memcached\_servers={%- for member in \_data.cache.members %}{{ member.host }}:11211{% if not loop.last %},{% endif %}{%- endfor %}

# In order to prevent excessive effort spent validating tokens, the middleware

# caches previously-seen tokens for a configurable duration (in seconds). Set

# to -1 to disable caching completely. (integer value)

#token\_cache\_time = 300

# DEPRECATED: Determines the frequency at which the list of revoked tokens is

# retrieved from the Identity service (in seconds). A high number of revocation

# events combined with a low cache duration may significantly reduce

# performance. Only valid for PKI tokens. This option has been deprecated in

# the Ocata release and will be removed in the P release. (integer value)

# This option is deprecated for removal since Ocata.

# Its value may be silently ignored in the future.

# Reason: PKI token format is no longer supported.

#revocation\_cache\_time = 10

{%- if \_data.cache.get('security', {}).get('enabled', False) %}

# (Optional) If defined, indicate whether token data should be authenticated or

# authenticated and encrypted. If MAC, token data is authenticated (with HMAC)

# in the cache. If ENCRYPT, token data is encrypted and authenticated in the

# cache. If the value is not one of these options or empty, auth\_token will

# raise an exception on initialization. (string value)

# Possible values:

# None - <No description provided>

# MAC - <No description provided>

# ENCRYPT - <No description provided>

#memcache\_security\_strategy = None

memcache\_security\_strategy = {{ \_data.cache.security.get('strategy', 'ENCRYPT') }}

# (Optional, mandatory if memcache\_security\_strategy is defined) This string is

# used for key derivation. (string value)

#memcache\_secret\_key = <None>

{%- if \_data.cache.security.secret\_key is not defined or not \_data.cache.security.secret\_key %}

{%- do salt.test.exception('cache.security.secret\_key is not defined: Please add secret\_key') %}

{%- else %}

memcache\_secret\_key = {{ \_data.cache.security.secret\_key }}

{%- endif %}

{%- endif %}

{%- endif %}

# (Optional) Number of seconds memcached server is considered dead before it is

# tried again. (integer value)

#memcache\_pool\_dead\_retry = 300

# (Optional) Maximum total number of open connections to every memcached

# server. (integer value)

#memcache\_pool\_maxsize = 10

# (Optional) Socket timeout in seconds for communicating with a memcached

# server. (integer value)

#memcache\_pool\_socket\_timeout = 3

# (Optional) Number of seconds a connection to memcached is held unused in the

# pool before it is closed. (integer value)

#memcache\_pool\_unused\_timeout = 60

# (Optional) Number of seconds that an operation will wait to get a memcached

# client connection from the pool. (integer value)

#memcache\_pool\_conn\_get\_timeout = 10

# (Optional) Use the advanced (eventlet safe) memcached client pool. The

# advanced pool will only work under python 2.x. (boolean value)

#memcache\_use\_advanced\_pool = false

# (Optional) Indicate whether to set the X-Service-Catalog header. If False,

# middleware will not ask for service catalog on token validation and will not

# set the X-Service-Catalog header. (boolean value)

#include\_service\_catalog = true

# Used to control the use and type of token binding. Can be set to: "disabled"

# to not check token binding. "permissive" (default) to validate binding

# information if the bind type is of a form known to the server and ignore it

# if not. "strict" like "permissive" but if the bind type is unknown the token

# will be rejected. "required" any form of token binding is needed to be

# allowed. Finally the name of a binding method that must be present in tokens.

# (string value)

#enforce\_token\_bind = permissive

# DEPRECATED: If true, the revocation list will be checked for cached tokens.

# This requires that PKI tokens are configured on the identity server. (boolean

# value)

# This option is deprecated for removal since Ocata.

# Its value may be silently ignored in the future.

# Reason: PKI token format is no longer supported.

#check\_revocations\_for\_cached = false

# DEPRECATED: Hash algorithms to use for hashing PKI tokens. This may be a

# single algorithm or multiple. The algorithms are those supported by Python

# standard hashlib.new(). The hashes will be tried in the order given, so put

# the preferred one first for performance. The result of the first hash will be

# stored in the cache. This will typically be set to multiple values only while

# migrating from a less secure algorithm to a more secure one. Once all the old

# tokens are expired this option should be set to a single value for better

# performance. (list value)

# This option is deprecated for removal since Ocata.

# Its value may be silently ignored in the future.

# Reason: PKI token format is no longer supported.

#hash\_algorithms = md5

# A choice of roles that must be present in a service token. Service tokens are

# allowed to request that an expired token can be used and so this check should

# tightly control that only actual services should be sending this token. Roles

# here are applied as an ANY check so any role in this list must be present.

# For backwards compatibility reasons this currently only affects the

# allow\_expired check. (list value)

#service\_token\_roles = service

# For backwards compatibility reasons we must let valid service tokens pass

# that don't pass the service\_token\_roles check as valid. Setting this true

# will become the default in a future release and should be enabled if

# possible. (boolean value)

#service\_token\_roles\_required = false

# Authentication type to load (string value)

# Deprecated group/name - [keystone\_authtoken]/auth\_plugin

#auth\_type = <None>

{%- if \_data.get('auth\_type', 'password') %}

auth\_type = {{ \_data.get('auth\_type', 'password') }}

{%- endif %}

# Config Section from which to load plugin specific options (string value)

#auth\_section = <None>