
python-route53 Documentation

Release 1.0

Greg Taylor

Apr 19, 2017

Contents

1 Assorted Info	3
2 User Guide	5
2.1 Installation	5
2.2 Quickstart	6
3 Reference	9
3.1 route53 API Reference	9
4 Indices and tables	23
Python Module Index	25

Amazon's [Route 53](#) is an excellent DNS service, with great stability and performance world-wide. The entire feature set is exposed through a web-based API. However, said API is somewhat difficult to work with in some cases.

python-route53 aims to provide the simplest possible API for [Route 53](#). It is also currently the only publicly available module with support for Python 3 (also supports 2.7).

License [BSD License](#)

CHAPTER 1

Assorted Info

- [Issue tracker](#) - Report bugs here.
- [GitHub project](#) - Source code lives here.
- [@gctaylor Twitter](#) - Tweets from the maintainer.
- [AWS's Official Route 53 Documentation](#) - Behaviors, specifics, more details.

Installation

Before installing `python-route53`, make sure that you have Python 2.7 or Python 3.x installed. If you do not have either of these, you'll need to use another module (`boto` is highly recommended).

If you're all set with Python 2.7 or Python 3, there are a few ways to install `python-route`.

Distribute & Pip

Installing `python-route53` is simple with `pip`:

```
pip install route53
```

`easy_install` works, too:

```
easy_install route53
```

Get the source

`python-route53` is developed on [GitHub](#) in the `python-route53` project.

You can either clone the repository:

```
git clone git://github.com/gtaylor/python-route53.git
```

Download the tarball:

```
curl -OL https://github.com/gtaylor/python-route53/tarball/master
```

Or download the zip:

```
curl -OL https://github.com/gtaylor/python-route53/zipball/master
```

Quickstart

This section goes over how to get up and running quickly. We'll assume that you have already followed the *Installation* instructions, and are ready to go.

Tip: It's best to combine our documentation with the [Route 53 Documentation](#). While we'll do our best to make this as simple as possible, it may be necessary to look at what they've got for more details on behavior and how things work.

AWS credentials

Before you can make your first query to [Route 53](#), you'll need obtain your API credentials. Visit your [security credentials](#) page and note your *Access Key ID* and *Secret Access Key*.

Instantiate the API client

Next, you'll want to import the module:

```
import route53
```

You can then instantiate a connection to Route53 via `route53.connect()`:

```
conn = route53.connect(
    aws_access_key_id='YOURACCESSKEYHERE',
    aws_secret_access_key='YOURSECRETACCESSKEYHERE',
)
```

You are now ready to roll. Continue reading to see how much fun there is to be had (hooray!).

Listing Hosted Zones

Let's say you want to retrieve a representation of all of your currently existing hosted zones. These roughly correspond to domains, ala `angry-squirrel.com`, or `python.org`.

The `Route53Connection.list_hosted_zones` method returns a generator of `HostedZone` instances:

```
# This is a generator.
for zone in conn.list_hosted_zones():
    # You can then do various things to the zone.
    print(zone.name)

    # Perhaps you want to see the record sets under this zone
    for record_set in zone.record_sets:
        print(record_set)

    # Or maybe you don't like this zone, and want to blow it away.
    zone.delete()
```

Creating a Hosted Zone

The `Route53Connection.create_hosted_zone` method creates hosted zones, and returns a tuple that contains a `HostedZone` instance, and some details about the pending change from the API:

```
new_zone, change_info = conn.create_hosted_zone(
    'some-domain.com.', comment='An optional comment.'
)

# You can then manipulate the HostedZone.
print("Zone ID", new_zone.id)

# This has some details about the change from the API.
print(change_info)
```

Note: Notice that we passed in a fully-qualified domain name, `some-domain.com.`, ending in a period.

In this case, `new_zone` is a new `HostedZone` instance, and `change_info` is a dict with some details about the changes pending (from the Route 53 API).

Retrieving a Hosted Zone

The `Route53Connection.get_hosted_zone_by_id` method retrieves a specific hosted zone, by Zone ID:

```
zone = conn.get_hosted_zone_by_id('ZONE-ID-HERE')
```

Note: A Zone ID is not the same thing as the domain name. The Zone ID is a unique string identifier for the hosted zone, as per Route 53's records.

Deleting a Hosted Zone

Simply call the `HostedZone.delete` method on a `HostedZone` to delete it:

```
zone = conn.get_hosted_zone_by_id('ZONE-ID-HERE')
zone.delete()
```

If you have record sets under the hosted zone, you'll need to delete those first, or an exception will be raised. Alternatively, you can call `delete()` with `force=True` to delete the record sets and the hosted zones:

```
zone.delete(force=True)
```

Creating a record set

Depending on which kind of record set you'd like to create, choose the appropriate `create_*_record` method on `HostedZone`. The methods return one of the `ResourceRecordSet` sub-classes:

```
new_record, change_info = zone.create_a_record(
    # Notice that this is a full-qualified name.
    name='test.some-domain.com.',
```

```
# A list of IP address entries, in the case fo an A record.
values=['8.8.8.8'],
)

# Or maybe we want a weighted round-robin set.
wrr_record1, change_info = zone.create_a_record(
    name='wrrtest.some-domain.com.',
    values=['8.8.8.8'],
    weight=1
    set_identifier='set123,
)
wrr_record2, change_info = zone.create_a_record(
    name='wrrtest.some-domain.com.',
    values=['6.6.6.6'],
    weight=2
    set_identifier='set123,
)
```

Listing record sets

In order to list record sets, use the `HostedZone.record_sets` property on `HostedZone`. Note that we don't currently implement any convenience methods for finding record sets, so this is the way to go:

```
# Note that this is a fully-qualified domain name.
name_to_match = 'fuzzy.bunny.com.'
for record_set in zone.record_sets:
    if record_set.name == name_to_match:
        print(record_set)
        # Stopping early may save some additional HTTP requests,
        # since zone.record_sets is a generator.
        break
```

While it may seem like extra work to craft these filters yourself, it does prevent needless additional iteration, and keeps the API more concise.

Changing a record set

Simply change one of the attributes on the various `ResourceRecordSet` sub-classed instances and call its `save()` method:

```
record_set.values = ['8.8.8.7']
record_set.save()
```

Deleting a record set

Similarly, delete a record set via its `delete()` method:

```
record_set.delete()
```

route53 API Reference

Below you will find the entire publicly exposed API for python-route53 documented in its entirety. If you find anything missing, lacking detail, or incorrect, please file an issue on the [issue tracker](#).

route53

The top-level route53 module is used as the default entry point to python-route53's functionality. You'll want to go through the `connect()` function to get a `Route53Connection` instance to work with the Route 53 API.

`route53.connect(aws_access_key_id=None, aws_secret_access_key=None, **kwargs)`

Instantiates and returns a `route53.connection.Route53Connection` instance, which is how you'll start your interactions with the Route 53 API.

Parameters

- `aws_access_key_id` (*str*) – Your AWS Access Key ID
- `aws_secret_access_key` (*str*) – Your AWS Secret Access Key

Return type `route53.connection.Route53Connection`

Returns A connection to Amazon's Route 53

route53.connection

class `route53.connection.Route53Connection(aws_access_key_id, aws_secret_access_key)`

Instances of this class are instantiated by the top-level `route53.connect()` function, and serve as a high level gateway to the Route 53 API. The majority of your interaction with these instances will probably be creating, deleting, and retrieving `HostedZone` instances.

Warning: Do not instantiate instances of this class yourself.

create_hosted_zone (*name*, *caller_reference=None*, *comment=None*)

Creates and returns a new hosted zone. Once a hosted zone is created, its details can't be changed.

Parameters

- **name** (*str*) – The name of the hosted zone to create.
- **caller_reference** (*str*) – A unique string that identifies the request and that allows failed `create_hosted_zone` requests to be retried without the risk of executing the operation twice. If no value is given, we'll generate a Type 4 UUID for you.
- **comment** (*str*) – An optional comment to attach to the zone.

Return type tuple**Returns** A tuple in the form of (`hosted_zone`, `change_info`). The `hosted_zone` variable contains a `HostedZone` instance matching the newly created zone, and `change_info` is a dict with some details about the API request.**delete_hosted_zone_by_id** (*id*)

Deletes a hosted zone, by hosted zone ID (not name).

Tip: For most cases, we recommend deleting hosted zones via a `HostedZone` instance's `HostedZone.delete` method, but this saves an HTTP request if you already know the zone's ID.

Note: Unlike `HostedZone.delete`, this method has no optional `force` kwarg.

Parameters **id** (*str*) – The hosted zone's ID (a short hash string).**Return type** dict**Returns** A dict of change info, which contains some details about the request.**endpoint_version** = '2012-02-29'**get_hosted_zone_by_id** (*id*)

Retrieves a hosted zone, by hosted zone ID (not name).

Parameters **id** (*str*) – The hosted zone's ID (a short hash string).**Return type** `HostedZone`**Returns** An `HostedZone` instance representing the requested hosted zone.**list_hosted_zones** (*page_chunks=100*)

List all hosted zones associated with this connection's account. Since this method returns a generator, you can pull as many or as few entries as you'd like, without having to query and receive every hosted zone you may have.

Parameters **page_chunks** (*int*) – This API call is “paginated” behind-the-scenes in order to break up large result sets. This number determines the maximum number of `HostedZone` instances to retrieve per request. The default is fine for almost everyone.**Return type** generator**Returns** A generator of `HostedZone` instances.

route53.hosted_zone

class route53.hosted_zone.**HostedZone**(*connection, id, name, caller_reference, resource_record_set_count, comment*)

A hosted zone is a collection of resource record sets hosted by Route 53. Like a traditional DNS zone file, a hosted zone represents a collection of resource record sets that are managed together under a single domain name. Each hosted zone has its own metadata and configuration information.

Warning: Do not instantiate this directly yourself. Go through one of the methods on `route53.connection.Route53Connection`.

create_a_record(*name, values, ttl=60, weight=None, region=None, set_identifier=None, alias_hosted_zone_id=None, alias_dns_name=None*)

Creates and returns an A record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).
- **weight** (*int*) – *For weighted record sets only.* Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location. Ranges from 0-255.
- **region** (*str*) – *For latency-based record sets.* The Amazon EC2 region where the resource that is specified in this resource record set resides.
- **set_identifier** (*str*) – *For weighted and latency resource record sets only.* An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type. 1-128 chars.
- **alias_hosted_zone_id** (*str*) – Alias A records have this specified. It appears to be the hosted zone ID for the ELB the Alias points at.
- **alias_dns_name** (*str*) – Alias A records have this specified. It is the DNS name for the ELB that the Alias points to.

Return type tuple

Returns A tuple in the form of (`rrset, change_info`), where `rrset` is the newly created `AResourceRecordSet` instance.

create_aaaa_record(*name, values, ttl=60, weight=None, region=None, set_identifier=None*)

Creates an AAAA record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).
- **weight** (*int*) – *For weighted record sets only.* Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location. Ranges from 0-255.
- **region** (*str*) – *For latency-based record sets.* The Amazon EC2 region where the resource that is specified in this resource record set resides.

- **set_identifier** (*str*) – For weighted and latency resource record sets only. An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type. 1-128 chars.

Return type tuple

Returns A tuple in the form of (*rrset*, *change_info*), where *rrset* is the newly created AAAAResourceRecordSet instance.

create_cname_record (*name*, *values*, *ttl=60*, *weight=None*, *region=None*, *set_identifier=None*)

Creates a CNAME record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).
- **weight** (*int*) – For weighted record sets only. Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location. Ranges from 0-255.
- **region** (*str*) – For latency-based record sets. The Amazon EC2 region where the resource that is specified in this resource record set resides.
- **set_identifier** (*str*) – For weighted and latency resource record sets only. An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type. 1-128 chars.

Return type tuple

Returns A tuple in the form of (*rrset*, *change_info*), where *rrset* is the newly created CNAMEResourceRecordSet instance.

create_mx_record (*name*, *values*, *ttl=60*)

Creates a MX record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).

Return type tuple

Returns A tuple in the form of (*rrset*, *change_info*), where *rrset* is the newly created MXResourceRecordSet instance.

create_ns_record (*name*, *values*, *ttl=60*)

Creates a NS record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).

Return type tuple

Returns A tuple in the form of (*rrset*, *change_info*), where *rrset* is the newly created NSResourceRecordSet instance.

create_ptr_record (*name, values, ttl=60*)

Creates a PTR record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).

Return type *tuple*

Returns A tuple in the form of (*rrset, change_info*), where *rrset* is the newly created PTRResourceRecordSet instance.

create_spf_record (*name, values, ttl=60*)

Creates a SPF record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).

Return type *tuple*

Returns A tuple in the form of (*rrset, change_info*), where *rrset* is the newly created SPFResourceRecordSet instance.

create_srv_record (*name, values, ttl=60*)

Creates a SRV record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).

Return type *tuple*

Returns A tuple in the form of (*rrset, change_info*), where *rrset* is the newly created SRVResourceRecordSet instance.

create_txt_record (*name, values, ttl=60, weight=None, region=None, set_identifier=None*)

Creates a TXT record attached to this hosted zone.

Parameters

- **name** (*str*) – The fully qualified name of the record to add.
- **values** (*list*) – A list of value strings for the record.
- **ttl** (*int*) – The time-to-live of the record (in seconds).
- **weight** (*int*) – *For weighted record sets only.* Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location. Ranges from 0-255.
- **region** (*str*) – *For latency-based record sets.* The Amazon EC2 region where the resource that is specified in this resource record set resides.

- **set_identifier** (*str*) – For weighted and latency resource record sets only. An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type. 1-128 chars.

Return type `tuple`

Returns A tuple in the form of (`rrset`, `change_info`), where `rrset` is the newly created `TXTResourceRecordSet` instance.

delete (*force=False*)

Deletes this hosted zone. After this method is ran, you won't be able to add records, or do anything else with the zone. You'd need to re-create it, as zones are read-only after creation.

Parameters **force** (*bool*) – If `True`, delete the `HostedZone`, even if it means nuking all associated record sets. If `False`, an exception is raised if this `HostedZone` has record sets.

Return type `dict`

Returns A dict of change info, which contains some details about the request.

nameservers

Return type `list`

Returns A list of nameserver strings for this hosted zone.

record_sets

Queries for the Resource Record Sets that are under this `HostedZone`. This is typically the way to go to find specific record sets, or to list them all.

We don't currently implement any filtering convenience method, since it is very easy to do this yourself, catered to your own needs. For example, if you find your match, you may choose to stop iterating on the generator, potentially saving yourself extra API queries (behind the scenes).

Warning: This result set can get pretty large if you have a ton of records.

Return type `generator`

Returns A generator of `ResourceRecordSet` sub-classes.

route53.resource_record_set

```
class route53.resource_record_set.AResourceRecordSet (alias_hosted_zone_id=None,
                                                       alias_dns_name=None, *args,
                                                       **kwargs)
```

Specific A record class. There are two kinds of A records:

- Regular A records.
- Alias A records. These point at an ELB instance instead of an IP.

Create these via `HostedZone.create_a_record`. Retrieve them via `HostedZone.record_sets`.

delete ()

Deletes this record set.

hosted_zone

Queries for this record set's `HostedZone`.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type *HostedZone*

Returns The matching *HostedZone* for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'A'

save()

Saves any changes to this record set.

class `route53.resource_record_set.AAAAResourceRecordSet` (*alias_hosted_zone_id=None*,
alias_dns_name=None,
**args, **kwargs*)

Specific AAAA record class. Create these via *HostedZone.create_aaaa_record*. Retrieve them via *HostedZone.record_sets*.

delete()

Deletes this record set.

hosted_zone

Queries for this record set's *HostedZone*.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type *HostedZone*

Returns The matching *HostedZone* for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'AAAA'

save ()
Saves any changes to this record set.

class route53.resource_record_set.**CNAMEResourceRecordSet** (*alias_hosted_zone_id=None,*
alias_dns_name=None,
**args, **kwargs*)

Specific CNAME record class. Create these via *HostedZone.create_cname_record*. Retrieve them via *HostedZone.record_sets*.

delete ()
Deletes this record set.

hosted_zone
Queries for this record set's HostedZone.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type *HostedZone*

Returns The matching *HostedZone* for this record set.

is_alias_record_set ()
Checks whether this is an A record in Alias mode.

Return type *bool*

Returns *True* if this is an A record in Alias mode, and *False* otherwise.

is_modified ()
Determines whether this record set has been modified since the last retrieval or save.

Return type *bool*

Returns *True* if the record set has been modified, and *False* if not.

rrset_type = 'CNAME'

save ()
Saves any changes to this record set.

class route53.resource_record_set.**MXResourceRecordSet** (*connection,* *zone_id,*
name, *ttl,* *records,*
weight=None, *region=None,*
set_identifier=None)

Specific MX record class. Create these via *HostedZone.create_mx_record*. Retrieve them via *HostedZone.record_sets*.

delete ()
Deletes this record set.

hosted_zone
Queries for this record set's HostedZone.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type *HostedZone*

Returns The matching *HostedZone* for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'MX'

save()

Saves any changes to this record set.

class `route53.resource_record_set.NSResourceRecordSet` (*connection*, *zone_id*,
name, *ttl*, *records*,
weight=None, *region=None*,
set_identifier=None)

Specific NS record class. Create these via `HostedZone.create_ns_record`. Retrieve them via `HostedZone.record_sets`.

delete()

Deletes this record set.

hosted_zone

Queries for this record set's `HostedZone`.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type `HostedZone`

Returns The matching `HostedZone` for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'NS'

save()

Saves any changes to this record set.

```
class route53.resource_record_set.PTRResourceRecordSet (connection, zone_id,
                                                         name, ttl, records,
                                                         weight=None, region=None,
                                                         set_identifier=None)
```

Specific PTR record class. Create these via `HostedZone.create_ptr_record`. Retrieve them via `HostedZone.record_sets`.

delete ()
Deletes this record set.

hosted_zone
Queries for this record set's `HostedZone`.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type `HostedZone`

Returns The matching `HostedZone` for this record set.

is_alias_record_set ()
Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified ()
Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'PTR'

save ()
Saves any changes to this record set.

```
class route53.resource_record_set.SOAResourceRecordSet (connection, zone_id,
                                                         name, ttl, records,
                                                         weight=None, region=None,
                                                         set_identifier=None)
```

Specific SOA record class. Retrieve these via `HostedZone.record_sets`. They can't be created.

delete ()
SOA records can't be created or deleted.

hosted_zone
Queries for this record set's `HostedZone`.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type `HostedZone`

Returns The matching `HostedZone` for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'SOA'

save()

Saves any changes to this record set.

```
class route53.resource_record_set.SPFResourceRecordSet (connection, zone_id,
                                                         name, ttl, records,
                                                         weight=None, region=None,
                                                         set_identifier=None)
```

Specific SPF record class. Create these via `HostedZone.create_spf_record`. Retrieve them via `HostedZone.record_sets`.

delete()

Deletes this record set.

hosted_zone

Queries for this record set's `HostedZone`.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type `HostedZone`

Returns The matching `HostedZone` for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'SPF'

save()

Saves any changes to this record set.

```
class route53.resource_record_set.SRVResourceRecordSet (connection, zone_id,
                                                         name, ttl, records,
                                                         weight=None, region=None,
                                                         set_identifier=None)
```

Specific SRV record class. Create these via `HostedZone.create_srv_record`. Retrieve them via `HostedZone.record_sets`.

delete ()
Deletes this record set.

hosted_zone
Queries for this record set's HostedZone.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type `HostedZone`

Returns The matching `HostedZone` for this record set.

is_alias_record_set ()
Checks whether this is an A record in Alias mode.

Return type `bool`

Returns `True` if this is an A record in Alias mode, and `False` otherwise.

is_modified ()
Determines whether this record set has been modified since the last retrieval or save.

Return type `bool`

Returns `True` if the record set has been modified, and `False` if not.

rrset_type = 'SRV'

save ()
Saves any changes to this record set.

```
class route53.resource_record_set.TXTResourceRecordSet (connection, zone_id,
                                                         name, ttl, records,
                                                         weight=None, region=None,
                                                         set_identifier=None)
```

Specific TXT record class. Create these via `HostedZone.create_txt_record`. Retrieve them via `HostedZone.record_sets`.

delete ()
Deletes this record set.

hosted_zone
Queries for this record set's HostedZone.

Note: This is not cached, it will always return the latest data from the Route 53 API.

Return type `HostedZone`

Returns The matching `HostedZone` for this record set.

is_alias_record_set()

Checks whether this is an A record in Alias mode.

Return type bool

Returns True if this is an A record in Alias mode, and False otherwise.

is_modified()

Determines whether this record set has been modified since the last retrieval or save.

Return type bool

Returns True if the record set has been modified, and False if not.

rrset_type = 'TXT'

save()

Saves any changes to this record set.

route53.exceptions

exception route53.exceptions.AlreadyDeletedError

Raised when the user tries to modify something on a hosted zone that has been deleted in Route53.

exception route53.exceptions.Route53Error

Base class for all Route53 API exceptions. Mostly here to allow end users to catch all Route53 exceptions.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

r

route53, 9
route53.connection, 9
route53.exceptions, 21
route53.hosted_zone, 11

A

AAAAResourceRecordSet (class
route53.resource_record_set), 15

AlreadyDeletedError, 21

AResourceRecordSet (class
route53.resource_record_set), 14

C

CNAMEResourceRecordSet (class
route53.resource_record_set), 16

connect() (in module route53), 9

create_a_record() (route53.hosted_zone.HostedZone
method), 11

create_aaaa_record() (route53.hosted_zone.HostedZone
method), 11

create_cname_record() (route53.hosted_zone.HostedZone
method), 12

create_hosted_zone() (route53.connection.Route53Connection
method), 10

create_mx_record() (route53.hosted_zone.HostedZone
method), 12

create_ns_record() (route53.hosted_zone.HostedZone
method), 12

create_ptr_record() (route53.hosted_zone.HostedZone
method), 12

create_spf_record() (route53.hosted_zone.HostedZone
method), 13

create_srv_record() (route53.hosted_zone.HostedZone
method), 13

create_txt_record() (route53.hosted_zone.HostedZone
method), 13

D

delete() (route53.hosted_zone.HostedZone method), 14

delete() (route53.resource_record_set.AAAAResourceRecordSet
method), 15

delete() (route53.resource_record_set.AResourceRecordSet
method), 14

delete() (route53.resource_record_set.CNAMEResourceRecordSet
method), 16

delete() (route53.resource_record_set.MXResourceRecordSet
method), 16

delete() (route53.resource_record_set.NSResourceRecordSet
method), 17

delete() (route53.resource_record_set.PTRResourceRecordSet
method), 18

delete() (route53.resource_record_set.SOAResourceRecordSet
method), 18

delete() (route53.resource_record_set.SPFResourceRecordSet
method), 19

delete() (route53.resource_record_set.SRVResourceRecordSet
method), 20

delete() (route53.resource_record_set.TXTResourceRecordSet
method), 20

delete_hosted_zone_by_id()
(route53.connection.Route53Connection
method), 10

E

endpoint_version (route53.connection.Route53Connection
attribute), 10

G

get_hosted_zone_by_id()
(route53.connection.Route53Connection
method), 10

H

hosted_zone (route53.resource_record_set.AAAAResourceRecordSet
attribute), 15

hosted_zone (route53.resource_record_set.AResourceRecordSet
attribute), 14

hosted_zone (route53.resource_record_set.CNAMEResourceRecordSet
attribute), 16

hosted_zone (route53.resource_record_set.MXResourceRecordSet
attribute), 16

hosted_zone (route53.resource_record_set.NSResourceRecordSet
attribute), 17

save() (route53.resource_record_set.CNAMEResourceRecordSet
method), 16

save() (route53.resource_record_set.MXResourceRecordSet
method), 17

save() (route53.resource_record_set.NSResourceRecordSet
method), 17

save() (route53.resource_record_set.PTRResourceRecordSet
method), 18

save() (route53.resource_record_set.SOAResourceRecordSet
method), 19

save() (route53.resource_record_set.SPFResourceRecordSet
method), 19

save() (route53.resource_record_set.SRVResourceRecordSet
method), 20

save() (route53.resource_record_set.TXTResourceRecordSet
method), 21

SOAResourceRecordSet (class in
route53.resource_record_set), 18

SPFResourceRecordSet (class in
route53.resource_record_set), 19

SRVResourceRecordSet (class in
route53.resource_record_set), 19

T

TXTResourceRecordSet (class in
route53.resource_record_set), 20