

# Package ‘aws.ses’

December 20, 2016

**Type** Package

**Title** AWS SES Client Package

**Version** 0.1.4

**Date** 2016-12-20

**Description** A simple client package for the Amazon Web Services (AWS) Simple Email Service (SES) <<http://aws.amazon.com/ses/>> REST API.

**License** GPL (>= 2)

**Imports** httr, jsonlite, aws.signature (>= 0.2.6)

**Suggests** aws.sns

**URL** <https://github.com/cloudyr/aws.ses>

**BugReports** <https://github.com/cloudyr/aws.ses/issues>

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Thomas J. Leeper [aut, cre]

**Maintainer** Thomas J. Leeper <[thosjleeper@gmail.com](mailto:thosjleeper@gmail.com)>

**Repository** CRAN

**Date/Publication** 2016-12-20 12:01:05

## R topics documented:

aws.ses-package . . . . .	2
get_dkim . . . . .	2
get_id_notification . . . . .	3
list_ids . . . . .	4
send_email . . . . .	5
sesPOST . . . . .	7
set_feedback . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

aws.ses-package      *aws.ses*

---

### Description

AWS SES Client Package

### Details

A simple client package for the Amazon Web Services (AWS) Simple Email Service (SES) REST API.

### Author(s)

Thomas J. Leeper <thosjleeper@gmail.com>

---

get\_dkim      *DKIM*

---

### Description

Manage DKIM

### Usage

```
get_dkim(identity, ...)  
  
set_dkim(identity, enabled = TRUE, ...)  
  
verify_dkim(domain, ...)
```

### Arguments

identity	An SES identity.
enabled	A logical.
domain	A character string containing a domain.
...	Additional arguments passed to <a href="#">sesPOST</a> .

### Examples

```
## Not run:  
verify_dkim("example.com")  
get_dkim("me@example.com")  
set_dkim("me@example.com", TRUE)  
get_dkim("me@example.com")  
  
## End(Not run)
```

---

get\_id\_notification    *Get/Set Notifications*

---

## Description

Get/Set ID Notifications

## Usage

```
get_id_notification(identity, ...)
```

```
set_id_notification(identity, type = c("Bounce", "Complaint", "Delivery"),  
  topic, ...)
```

## Arguments

identity	An SES identity.
type	A character string specifying a notification type.
topic	An SNS topic name. See <a href="#">create_topic</a>
...	Additional arguments passed to <a href="#">sesPOST</a> .

## See Also

[verify\\_id](#), [create\\_topic](#)

## Examples

```
## Not run:  
# get  
get_id_notification("example@example.com")  
  
# set  
if (require("aws.sns")) {  
  top <- create_topic("ses_email_bounce")  
  set_id_notification("example@example.com", "Bounce", top)  
  get_id_notification("example@example.com")  
  
  # cleanup  
  delete_topic(top)  
}  
  
## End(Not run)
```

---

list_ids	<i>SES Identities</i>
----------	-----------------------

---

### Description

Manage SES Identities

### Usage

```
list_ids(type = c("EmailAddress", "Domain"), nmax = NULL,  
         next_token = NULL, ...)
```

```
verify_id(address, domain, ...)
```

```
get_verification_attrs(identity, ...)
```

```
delete_id(identity, ...)
```

### Arguments

type	A character string specifying the identity type.
nmax	An integer specifying the maximum number of identities to return.
next_token	A pagination token
address	A character string specifying an email address.
domain	A character string specifying a domain.
identity	An SES identity.
...	Additional arguments passed to <a href="#">sesPOST</a> .

### Details

`verify_id` sends an email verification request to the specified email address.

### Value

`list_ids` returns a character vector of verified email addresses or domains.

### See Also

[get\\_id\\_notification](#)

**Examples**

```
## Not run:
# verify an address
verify_id("example@example.com")
get_verification_attrs("example@example.com")

list_ids()

# remove identity
delete_id("example@example.com")

## End(Not run)
```

---

send\_email

*Send Email via SES*


---

**Description**

Send email via Amazon SES service

**Usage**

```
send_email(message, html, raw = NULL, subject, from, to = NULL, cc = NULL,
  bcc = NULL, replyto = NULL, returnpath = NULL, charset.subject = NULL,
  charset.message = NULL, charset.html = NULL, ...)

get_quota(...)

get_statistics(...)
```

**Arguments**

message	A character string specifying the message body (as plain text). Must specify html, message, or both; or alternatively raw.
html	A character string specifying the message body (as HTML). Must specify html, message, or both; or alternatively raw.
raw	A raw vector containing a valid email body (of no more than 10MB in size). Can only specify raw xor message/html.
subject	A character string specifying the email subject
from	A character string specifying the sender
to	A character vector of TO recipient email addresses.
cc	A character vector of CC recipient email addresses.
bcc	A character vector of BCC recipient email addresses.
replyto	A character vector of reply-to email addresses.

returnpath	A character string specifying the email address to which feedback and bounces are sent.
charset.subject	An optional character string specifying the character set, e.g., “UTF-8”, “ISO-8859-1”, etc. if subject is not ASCII.
charset.message	An optional character string specifying the character set, e.g., “UTF-8”, “ISO-8859-1”, etc. if message is not ASCII.
charset.html	An optional character string specifying the character set, e.g., “UTF-8”, “ISO-8859-1”, etc. if html is not ASCII.
...	Additional arguments passed to <code>sesPOST</code> .

### Details

Send a test or raw email message. `get_quota` and `get_statistics` provide information on remaining and used email rate limits, respectively.

### Value

A character string containing the message ID.

### Examples

```
## Not run:
# verify email address
verify_identity("me@example.com")

# if email verified, can be used to send a message
a <- get_verification_attrs("me@example.com")
if (a[[1]]$VerificationStatus == "Success") {
  # simple plain-text email
  send_email("Test Email Body", subject = "Test Email",
            from = "me@example.com", to = "recipient@example.com")

  # html and plain text versions
  send_email(message = "Plain text body",
            html = "<div><p style='font-weight:bold;'>HTML text body</p></div>",
            subject = "Test Email",
            from = "me@example.com", to = "recipient@example.com")
}

## End(Not run)
## Not run:
get_quota()
get_statistics()

## End(Not run)
```

---

 sesPOST

*SES HTTP Requests*


---

**Description**

Low-level SES POST function

**Usage**

```
sesPOST(query = list(), body = NULL,
        region = Sys.getenv("AWS_DEFAULT_REGION", "us-east-1"),
        key = Sys.getenv("AWS_ACCESS_KEY_ID"),
        secret = Sys.getenv("AWS_SECRET_ACCESS_KEY"),
        session_token = Sys.getenv("AWS_SESSION_TOKEN"), ...)
```

**Arguments**

query	A list containing query string parameters
body	A list of query-like parameters to be passed as a form-encoded message body.
region	A character string containing an AWS region. If missing, the default “us-east-1” is used.
key	A character string containing an AWS Access Key ID. The default is pulled from environment variable “AWS_ACCESS_KEY_ID”.
secret	A character string containing an AWS Secret Access Key. The default is pulled from environment variable “AWS_SECRET_ACCESS_KEY”.
session_token	Optionally, a character string containing an AWS temporary Session Token. If missing, defaults to value stored in environment variable “AWS_SESSION_TOKEN”.
...	Additional arguments passed to <a href="#">POST</a> .

---

set\_feedback

*Enable SES Feedback*


---

**Description**

Enable feedback forwarding

**Usage**

```
set_feedback(identity, forwarding = TRUE, ...)
```

**Arguments**

identity	An SES identity.
forwarding	A logical indicating whether to enable (TRUE) or disable (FALSE) feedback forwarding.
...	Additional arguments passed to <a href="#">sesPOST</a> .

# Index

## \*Topic **package**

aws.ses-package, 2

aws.ses (aws.ses-package), 2

aws.ses-package, 2

create\_topic, 3

delete\_id (list\_ids), 4

get\_dkim, 2

get\_id\_notification, 3, 4

get\_quota (send\_email), 5

get\_statistics (send\_email), 5

get\_verification\_attrs (list\_ids), 4

list\_ids, 4

POST, 7

send\_email, 5

sesPOST, 2–4, 6, 7, 7

set\_dkim (get\_dkim), 2

set\_feedback, 7

set\_id\_notification

(get\_id\_notification), 3

verify\_dkim (get\_dkim), 2

verify\_id, 3

verify\_id (list\_ids), 4