

Package ‘blastula’

August 21, 2017

Title Easily Send HTML Email Messages

Version 0.1

Description Compose and send out responsive HTML email messages that render perfectly across a range of email clients and device sizes. Messages are composed using 'Markdown' and a text interpolation system that allows for the injection of evaluated R code within the message body, footer, and subject line. Helper functions let the user insert embedded images, web link buttons, and 'ggplot2' plot objects into the message body. Messages can be sent through an 'SMTP' server or through the 'Mailgun' API service <<http://mailgun.com/>>.

Depends R (>= 3.4.0)

License MIT + file LICENSE

Imports commonmark (>= 1.2), ggplot2 (>= 2.2.1), glue (>= 1.1.1),
htmltools (>= 0.3.6), httr (>= 1.2.1), knitr (>= 1.16),
magrittr (>= 1.5), mailR (>= 0.4.1), rJava (>= 0.9-8), stringr
(>= 1.2.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests covr, testthat

NeedsCompilation no

Author Rich Iannone [aut, cre]

Maintainer Rich Iannone <riannone@me.com>

Repository CRAN

Date/Publication 2017-08-20 22:39:17 UTC

R topics documented:

add_cta_button	2
add_ggplot	3
add_image	4

compose_email	5
create_email_creds_file	7
preview_email	8
send_by_mailgun	8
send_email_out	9
%>%	10

Index 11

add_cta_button	<i>Helper function for adding a CTA button</i>
----------------	--

Description

Add a call to action button inside the body of the email with this helper function. There are options to specify the button text, the URL, and the button's alignment.

Usage

```
add_cta_button(url, text, align = "left")
```

Arguments

url	a URL for the button.
text	the text that is placed atop the CTA button.
align	the alignment of the button inside the main content area. Options are left (the default), right, and center.

Value

a character object with an HTML fragment that can be placed inside the message body wherever the CTA button should appear.

Examples

```
# Create the button as an HTML fragment
cta_button <-
  add_cta_button(
    url = "http://www.website.net",
    text = "Press This Button",
    align = "center")

# Include the button in the email
# message body by simply referencing
# the `cta_button` object
email <-
  compose_email(
    body = "
    Hello!
```

```
Below is a call. It's a call \\  
to action. Press it!
```

```
{cta_button}
```

```
Cheers  
")
```

add_ggplot*Helper function for adding an ggplot*

Description

Add an ggplot plot inside the body of the email with this helper function.

Usage

```
add_ggplot(plot_object, width, height)
```

Arguments

plot_object	the ggplot plot object.
width	the width of the output plot in inches.
height	the height of the output plot in inches.

Value

a character object with an HTML fragment that can be placed inside the message body wherever the plot image should appear.

Examples

```
library(ggplot2)  
  
# Create a ggplot plot  
plot <-  
  ggplot(  
    data = mtcars,  
    aes(x = disp, y = hp,  
        color = wt, size = mpg)) +  
    geom_point()  
  
# Create an HTML fragment that  
# contains an the ggplot as an  
# embedded plot  
plot_html <-  
  add_ggplot(  
    plot_object = plot,
```

```
      height = 5,
      width = 7)

# Include the plot in the email
# message body by simply referencing
# the `plot_html` object
email <-
  compose_email(
    body = "
    Hello!

    Here is a very important plot \\
    that will change the way you \\
    look at cars forever.

    {plot_html}

    So useful, right?
    ") %>% preview_email()
```

add_image

Helper function for adding an image

Description

Add an image inside the body of the email with this helper function.

Usage

```
add_image(file)
```

Arguments

file a path to an image file.

Value

a character object with an HTML fragment that can be placed inside the message body wherever the image should appear.

Examples

```
# Create an HTML fragment that
# contains an image
img_file_path <-
  system.file(
    "graphics",
    "melon_cat.png",
    package = "blastula")
```

```
img_file_html <-
  add_image(
    file = img_file_path)

# Include the image in the email
# message body by simply referencing
# the `img_file_html` object
email <-
  compose_email(
    body = "
    Hello!

    Take a look at this image:

    {img_file_html}

    Funny, right?
  ")
```

compose_email

Create the email message

Description

Create an email message. String interpolation is possible for the text comprising the email body, footer, and preheader text. This is done by using curly braces to enclose R code chunks. Variables can be specified in the function call (using named arguments with `...`), and any variables not found in `...` will be searched for in the global environment.

Usage

```
compose_email(body = NULL, footer = NULL, .preheader_text = NULL,
  .title = NULL, ...)
```

Arguments

body	the main body of text for the email message. Markdown can be used here (along with string interpolation via curly braces and named arguments) to construct the main text.
footer	the footer text for the email message. As with the body, Markdown and string interpolation can be used here.
.preheader_text	text that appears before the subject in some email clients. This must be plaintext.
.title	the title of the email message. This is not the subject but the HTML title text which may appear in limited circumstances.
...	expression strings for string interpolation for the body, footer, and preheader_text string data.

Value

an `email_message` object, which can be used for previewing with the `preview_email()` function or for sending out actual emails with the `send_email_out()` function.

Examples

```
# Create a simple email message using
# Markdown formatting
email <-
  compose_email(
    body = "
    Hello!

    ## This a section heading

    We can use Markdown formatting \\
    to embolden text or to add \\
    *emphasis*. This is exciting, \\
    right?

    Cheers")

# The email message can always be
# previewed using `preview_email()`
preview_email(email = email)

# We can use string interpolation to
# add in R code or strings assigned
# to variables; variables can be
# obtained from the global workspace
# or from temporary variables in the
# function call
sender_name <- "Mike"

email <-
  compose_email(
    body = "
    Hello!

    I just wanted to let you \\
    know that the {thing} that \\
    asked me for is ready to \\
    pick up. So, come over and \\
    do that.

    Cheers,

    {sender_name}",
    thing = "report")
```

`create_email_creds_file`*Create a file with email access credentials*

Description

Creates a file with access credentials for the purpose of automatically emailing notification messages.

Usage

```
create_email_creds_file(file, sender, host, port, user, password,  
                        use_ssl = TRUE, use_tls = FALSE, authenticate = TRUE)
```

Arguments

<code>file</code>	a file path for the credentials file to be stored on disk.
<code>sender</code>	the sender name.
<code>host</code>	the host name.
<code>port</code>	the port number.
<code>user</code>	the username for the email account.
<code>password</code>	the password associated with the user's email address.
<code>use_ssl</code>	an option as to whether to use SSL; supply a TRUE or FALSE value (TRUE is the default value).
<code>use_tls</code>	a logical value to indicate whether to use TLS.
<code>authenticate</code>	an option as to whether to authenticate; supply a TRUE or FALSE value (TRUE is the default value).

Examples

```
## Not run:  
# Create a credentials file to facilitate  
# the sending of email messages  
create_email_creds_file(  
  file = "~/email_file",  
  sender = "correspondences@blastula.org",  
  host = "smtp.blastula.org",  
  port = 465,  
  user = "have_a@blastula.org",  
  password = "*****")  
  
## End(Not run)
```

preview_email *Preview an email message*

Description

Preview an HTML email before sending it out.

Usage

```
preview_email(email)
```

Arguments

email an email_message object.

send_by_mailgun *Send an email message through the Mailgun API*

Description

Send an email message via the Mailgun API. This requires an account with Mailgun.

Usage

```
send_by_mailgun(message, subject = NULL, from, recipients, url, api_key)
```

Arguments

message the email message object, as created by the compose_email() function. The object's class is email_message

subject the subject of the email.

from the email address of the sender. This does not have to be the same email that is associated with the account actually sending the message.

recipients a vector of email addresses.

url the URL for the sending domain.

api_key the API key registered to the mailgun service.

Examples

```
## Not run:
# Create a simple email message using
# Markdown formatting
email <-
  compose_email(
    body = "
    Hello!

    ## This a section heading

    We can use Markdown formatting \\
    to embolden text or to add \\
    *emphasis*. This is exciting, \\
    right?

    Cheers")

# Generate a vector of recipients
recipient_list <-
  c("person_1@site.net",
    "person_2@site.net")

# Send it to multiple people through
# the Mailgun API
email %>%
  send_by_mailgun(
    subject = "Sent through Mailgun",
    from = "The Sender <sender@send.org>",
    recipients = recipient_list,
    url = "<..mailgun_sending_domain..>",
    api = "<..mailgun_api_key..>")

## End(Not run)
```

send_email_out

Send an email message

Description

Send an email message to one or more recipients.

Usage

```
send_email_out(message, subject = NULL, from = NULL, recipients = NULL,
  creds_file = NULL, sender = NULL, host = NULL, port = NULL,
  user = NULL, password = NULL, use_ssl = TRUE, use_tls = FALSE,
  authenticate = TRUE, debug = FALSE)
```

Arguments

message	the email message object, as created by the <code>compose_email()</code> function. The object's class is <code>email_message</code>
subject	the subject of the email.
from	the email address of the sender. This does not have to be the same email that is associated with the account actually sending the message.
recipients	a vector of email addresses.
creds_file	an optional path to an email credentials file. This file must be created by the <code>create_email_creds_file</code> function.
sender	the sender name.
host	the email host.
port	the port associated with the email account.
user	the username associated with the email account.
password	the password associated with the email account.
use_ssl	a logical value to indicate whether to use SSL.
use_tls	a logical value to indicate whether to use TLS.
authenticate	a logical value to indicate whether to use authentication.
debug	a logical value to indicate whether a detailed debug information should be printed to the console during sending of email.

%>%

The magrittr pipe

Description

`blastula` uses the pipe function, `%>%` to turn function composition into a series of imperative statements.

Index

`%>%`, [10](#)

`add_cta_button`, [2](#)

`add_ggplot`, [3](#)

`add_image`, [4](#)

`compose_email`, [5](#)

`create_email_creds_file`, [7](#)

`preview_email`, [8](#)

`send_by_mailgun`, [8](#)

`send_email_out`, [9](#)