

# Sending Mail

The mail API provides two ways to send an email message: the [mail.send\\_mail\(\)](#) function and the [EmailMessage](#) class.

Sending is asynchronous: the `mail.send_mail()` function and the `EmailMessage.send()` method transmit the message data to the mail service, then return. The mail service queues the message, then attempts to send it, retrying if the destination mail server is unavailable. Errors and bounce messages are sent to the sender address for the email message.

For information on who can send mail and for guidelines on sending bulk mail, see the [Mail API Overview](#).

## Sending mail with `mail.send_mail()`

---

To send mail using the [mail.send\\_mail\(\)](#) function, use the fields of the email message as parameters, including the sender, the recipients, the subject and the body of the message. For example:

[appengine/standard/mail/send\\_mail.py](#)

[VIEW ON GITHUB](#)

## Sending mail with `EmailMessage`

---

To send mail using objects with the [EmailMessage](#) class, pass the fields of the email message to the `EmailMessage` constructor and use attributes of the instance to update message.

The `send()` method sends the email message represented by the instance's attributes. An application can re-use an `EmailMessage` instance by modifying attributes and calling the `send()` method again.

[appengine/standard/mail/send\\_message.py](#)

[VIEW ON GITHUB](#)

Here's an example of sending a message to confirm an email address:

```
class UserSignupHandler(webapp2.RequestHandler):
    """Serves the email address sign up form."""

    def post(self):
```

```

        user_address = self.request.get('email_address')

        if not mail.is_email_valid(user_address):
            self.get() # Show the form again.
        else:
            confirmation_url =
create_new_user_confirmation(user_address)
            sender_address = (
                'Example.com Support
<{}@appspot.gserviceaccount.com>'.format(
                    app_identity.get_application_id()))
            subject = 'Confirm your registration'
            body = """Thank you for creating an account!
Please confirm your email address by clicking on the link below:

{}
""".format(confirmation_url)
            mail.send_mail(sender_address, user_address, subject,
body)

```