

Consider the following function that sends an email:

```
def send_email(
    message: str, to_address: str, from_address: str = DEFAULT_EMAIL
) -> None:
    server = SMTP()
    server.connect(HOST, PORT)
    server.login(LOGIN, PASSWORD)
    server.sendmail(from_address, to_address, message)
    server.quit()
```

The `send_email` function depends on the `SMTP` type from `smtpplib`. Removing the dependency here is a bit more complicated because at the moment, `send_email` creates an instance of `SMTP` directly. So the first step is to change this so that `send_email` now relies on dependency injection:

```
def send_email(
    server: SMTP,
    message: str, to_address: str, from_address: str = DEFAULT_EMAIL
) -> None:
    server.connect(HOST, PORT)
    # etc
```

Now that the function no longer creates the object, we can add a Protocol class that defines the interface of an SMTP server and use that instead of the `SMTP` type. For the full solution, see [solution_2.py](#).