# PHPMailer code examples

This folder contains a collection of examples of using [PHPMailer](https://github.com/PHPMailer/PHPMailer).

## About testing email sending

When working on email sending code you'll find yourself worrying about what might happen if all these test emails got sent to your mailing list. The solution is to use a fake mail server, one that acts just like the real thing, but just doesn't actually send anything out. Some offer web interfaces, feedback, logging, the ability to return specific error codes, all things that are useful for testing error handling, authentication etc. Here's a selection of mail testing tools you might like to try:

* [FakeSMTP](https://github.com/Nilhcem/FakeSMTP), a Java desktop app with the ability to show an SMTP log and save messages to a folder.
* [FakeEmail](https://github.com/isotoma/FakeEmail), a Python-based fake mail server with a web interface.
* [smtp-sink](http://www.postfix.org/smtp-sink.1.html), part of the Postfix mail server, so you probably already have this installed. This is used in the Travis-CI configuration to run PHPMailer's unit tests.
* [smtp4dev](http://smtp4dev.codeplex.com), a dummy SMTP server for Windows.
* [fakesendmail.sh](https://github.com/PHPMailer/PHPMailer/blob/master/test/fakesendmail.sh), part of PHPMailer's test setup, this is a shell script that emulates sendmail for testing 'mail' or 'sendmail' methods in PHPMailer.
* [msglint](http://tools.ietf.org/tools/msglint/), not a mail server, the IETF's MIME structure analyser checks the formatting of your messages.

Security note

Before running these examples you'll need to rename them with '.php' extensions. They are supplied as '.phps' files which will usually be displayed with syntax highlighting by PHP instead of running them. This prevents potential security issues with running potential spam-gateway code if you happen to deploy these code examples on a live site - *please don't do that!* Similarly, don't leave your passwords in these files as they will be visible to the world!

## [**code\_generator.phps**](http://docs.google.com/code_generator.phps)

This script is a simple code generator - fill in the form and hit submit, and it will use when you entered to email you a message, and will also generate PHP code using your settings that you can copy and paste to use in your own apps. If you need to get going quickly, this is probably the best place to start.

## [**mail.phps**](http://docs.google.com/mail.phps)

This script is a basic example which creates an email message from an external HTML file, creates a plain text body, sets various addresses, adds an attachment and sends the message. It uses PHP's built-in mail() function which is the simplest to use, but relies on the presence of a local mail server, something which is not usually available on Windows. If you find yourself in that situation, either install a local mail server, or use a remote one and send using SMTP instead.

## [**exceptions.phps**](http://docs.google.com/exceptions.phps)

The same as the mail example, but shows how to use PHPMailer's optional exceptions for error handling.

## [**smtp.phps**](http://docs.google.com/smtp.phps)

A simple example sending using SMTP with authentication.

## [**smtp\_no\_auth.phps**](http://docs.google.com/smtp_no_auth.phps)

A simple example sending using SMTP without authentication.

## [**sendmail.phps**](http://docs.google.com/sendmail.phps)

A simple example using sendmail. Sendmail is a program (usually found on Linux/BSD, OS X and other UNIX-alikes) that can be used to submit messages to a local mail server without a lengthy SMTP conversation. It's probably the fastest sending mechanism, but lacks some error reporting features. There are sendmail emulators for most popular mail servers including postfix, qmail, exim etc.

## [**gmail.phps**](http://docs.google.com/gmail.phps)

Submitting email via Google's Gmail service is a popular use of PHPMailer. It's much the same as normal SMTP sending, just with some specific settings, namely using TLS encryption, authentication is enabled, and it connects to the SMTP submission port 587 on the smtp.gmail.com host. This example does all that.

## [**pop\_before\_smtp.phps**](http://docs.google.com/pop_before_smtp.phps)

Before effective SMTP authentication mechanisms were available, it was common for ISPs to use POP-before-SMTP authentication. As it implies, you authenticate using the POP3 protocol (an older protocol now mostly replaced by the far superior IMAP), and then the SMTP server will allow send access from your IP address for a short while, usually 5-15 minutes. PHPMailer includes a POP3 protocol client, so it can carry out this sequence - it's just like a normal SMTP conversation (without authentication), but connects via POP first.

## [**mailing\_list.phps**](http://docs.google.com/mailing_list.phps)

This is a somewhat naïve example of sending similar emails to a list of different addresses. It sets up a PHPMailer instance using SMTP, then connects to a MySQL database to retrieve a list of recipients. The code loops over this list, sending email to each person using their info and marks them as sent in the database. It makes use of SMTP keepalive which saves reconnecting and re-authenticating between each message.

## [**smtp\_check.phps**](http://docs.google.com/smtp_check.phps)

This is an example showing how to use the SMTP class by itself (without PHPMailer) to check an SMTP connection.

Most of these examples use the 'example.com' domain. This domain is reserved by IANA for illustrative purposes, as documented in [RFC 2606](http://tools.ietf.org/html/rfc2606). Don't use made-up domains like 'mydomain.com' or 'somedomain.com' in examples as someone, somewhere, probably owns them!