**Forward:**

This is a very customized implementation of Exchange Web Services (2010 SP2) built for a specific customer to use with an existing application. The functions provided are by no means comprehensive and they a very tailored to the application. If you need to do something similar to what I’ve done here with jamesiarmes’ php-ews package (<https://github.com/jamesiarmes/php-ews>) I’d recommend reading his documentation, reading this <https://www.howtoforge.com/talking-soap-with-exchange>, and reading <https://thomas.rabaix.net/blog/2008/03/using-soap-php-with-ntlm-authentication> to get a better understanding of the object structure and the SOAP message structure. This is all to say, this code is probably too specific for what you’re looking to do, but it will probably serve as a decent example for how to implement this sort of thing. But, of course, you’re more than welcome to use it.

**Problem:**

There is a need to access, update, create, and delete the contact information stored on an exchange 2010 (service pack 2) server. More specifically, this information needs to be accessed from an application using Exchange Web Services and PHP. The contacts that need to be manipulated are being shared with a ‘dummy’ exchange account, which is given read/write access. The solution only needs to be able to manipulate “stock” contact fields.

**Exchange Interface Package:**

The exchange interface provides a simple functions package that allows application developers to manipulate exchange contacts. The exchange interface is built upon the php-ews package (<https://github.com/jamesiarmes/php-ews>) and shares all dependencies.

**File Hierarchy:**

* exchangeinterface.php
  + exchangecreatedeletefunctions.php
  + exchangeupdatefunctions.php
  + exchangeremovefunctions.php
  + exchangeutilities.php

**Configuration:**

To configure the exchange interface, one needs to edit the newWebService() function which is defined in exchangeutilities.php. Recall, that the credentials that you provide should be for the dummy account.

function newWebService(){

$server = 'yourserver.yourdomain.com';

$username = 'username';

$password = 'password';

….

}

Additionally, if users should find all occurrences of [masteraccount@domain.com](mailto:masteraccount@domain.com) and replace those with the email of the true owner of the contacts being shared with your dummy account.

Then simply include exchangeinterface.php and the rest is comes along with it.