

Open a PostalMethods developer account

Open a free PostalMethods developer account. All you're required to supply is your email address, a requested username and a password.

The free developer account will allow you to simulate the sending of letters through PostalMethods. You will get access to a fully functional control panel working in "Development Mode" (check out the top right corner of your control panel screen once you have access - you can see your mode at any time).

You can do almost everything that a live account can do, including:

- Submit messages for web-to-postal messaging via SMTP, XML Web Service, and HTTP POST.
- Review the image of a letter through the control panel.
- Check the status of a message through the control panel (we simulate a successful delivery).
- Query the status of a message via XML Web Service and HTTP POST.
- Set user preferences - define permissions, stationery types, defaults, etc.
- Turn your account into a live account in "Production Mode" by buying a prepaid package from within the control panel.

The only thing you can't do with a free developer account is to send a live letter.

Prepare your development environment

If you intend to submit your messages via SOAP, follow these steps to verify that your environment is capable of sending web-to-postal messages through PostalMethods:

1. From your server, verify that you can view a live page on the PostalMethods server at <https://api.postalmethods.com/PostalMain.aspx> (HTTPS, port 443) or at <http://api.postalmethods.com/PostalMain.aspx> (HTTP, port 80). If you can't, check your internet connection, firewall settings, and proxy settings. We recommend using the HTTPS secure connection.
2. Scroll down to your development environment for further instructions:
 - C# and VB.NET using Visual Studio 2008
 - C# and VB.NET using Visual Studio 2002 or 2005
 - Java
 - PHP5
 - Perl
 - Python
 - Ruby
 - ASP
 - ColdFusion

Java

3. You need to have Java 1.5 or newer to access the PostalMethods API using Java. To determine your version of Java, run following command on the command line:

```
java -version
```

The output will look something like:

```
java version "1.6.0_03"  
Java(TM) SE Runtime Environment (build 1.6.0_03-b05)  
Java HotSpot(TM) Client VM (build 1.6.0_03-b05, mixed mode, sharing)
```

You should also check to make sure that you have the Java development tools installed. You can do this by running the following command:

```
javac -version
```

The output will look something like this:

```
javac 1.6.0_06
```

4. Download and unpack the latest PostalMethods Java toolkit [files](#) into a directory. On Windows, use your favorite zip utility to unzip the file into its own directory. On Linux, run the command `unzip pmjava.zip`. You will now have a directory called *pmjava* that contains all of the files you need to get started.

Visual Studio 2002 & 2005 (C# and VB.NET)

3. Create a new project
4. Select the "Console Application" template
5. Create a Web Reference:
Using the Solution Explorer, right-click on the project name (by default: "ConsoleApplication1"):
 - o Click the "Add Web Reference" button
 - o In the URL line, write **https://api.postalmethods.com/PostalMain.asmx** and hit the "Go" button
 - o Once the Web Service is detected, click the "Add Reference" button

Visual Studio 2008 (C# and VB.NET)

3. Create a new project
4. Select the "Console Application" template
5. Create a Web Reference:
Using the Solution Explorer, right-click on the project name (by default: "ConsoleApplication1"):
 - o Select "Add Service Reference"
 - o Click the "Advanced" button
 - o Click the "Add Web Reference" button
 - o In the URL line, enter **https://api.postalmethods.com/PostalMain.asmx** and hit the "Go" button
 - o Once the Web Service is detected, click the "Add Reference" button

PHP 5

3. Determine your version of PHP by running a file with the following contents on your server:

```
4. <?php  
5. echo phpinfo();  
?>
```

The title of the resulting page will indicate the version number of your PHP installation, for example **PHP version 5.2**. Proceed if you have version 5 or higher (no support is offered here for PHP4.x)

6. Verify your SOAP library - in the resulting page from step #3, search for the phrase **--enable soap**. If it's present - you have the required SOAP library installed (it usually is). If not, you need to either recompile PHP with this option enabled, or use PostalMethods via HTTP POST or SMTP.

Perl

3. You need to have the SOAP::Lite module, version 0.60 or greater, installed. To verify that SOAP:Lite is properly installed and to display its version, run this code from a Perl script:

```
4. use SOAP::Lite;
   print $SOAP::Lite::VERSION . "\n";
```

5. If SOAP::Lite is not installed you will need to install the module by running:

```
perl -MCPAN -e 'install SOAP::Lite'
```

Python

3. Verify Python is installed. Invoke the Python interpreter by typing 'python' in a command shell. You should see something similar to this:

```
4. $ python
5. Python 2.5.2 (r252:60911, Apr 21 2008, 11:12:42) [GCC 4.2.3 (Ubuntu 4.2.3-
   2ubuntu7)] on linux2 Type "help", "copyright", "credits" or "license" for more
   information.
   >>>
```

6. Install the Zolera SOAP Infrastructure (ZSI) module. Verify that it is available by attempting to import it:

```
7. >>> import ZSI
   >>>
```

If you do not have ZSI you will need to install it. The steps for doing this differ from one system to another but the installation is described with the ZSI distribution.

Ruby

Unix

3. Make sure you have a working Ruby and gem service and that you have permissions to 'sudo'
4. In 'Terminal', run this command: "sudo gem install postalmethods"

Windows

3. Make sure your path includes the Ruby and gem command
4. Run the following command: "gem install postalmethods". You may use the "Ruby for Windows" installer.

ASP

3. You need to have Microsoft SOAP Toolkit Version 3.0 installed. To verify that it is properly installed, add the following code to an ASP page:

```
4. <%  
5. Dim X : Set X = Server.CreateObject("MSSOAP.SoapClient30")  
6. Response.Write "SOAP is available"  
%>
```

7. To run your script Using IIS:
 - a. Place your script in a folder (e.g. C:\PostalMethods\myscript.asp)
 - b. Open IIS Manager. Under Local computer -> Web sites -> Default web site -> [Right click] new -> Virtual Directory. Alias: *PostalMethods*; Directory: C:\PostalMethods
 - c. Open your browser and type: *http://localhost/interfax/myscript.asp*
8. If you are sending letters from behind a proxy using Microsoft SOAP Toolkit Version 3.0, you can use one of the two following options:

- o Option 1 - Use automatic detection by:

```
objSoap.ConnectorProperty("EnableAutoProxy") = True
```

- o Option 2 - Specify a proxy server:

```
objSoap.ConnectorProperty("ProxyServer") = "192.168.0.100"
```

- o Optionally, if the proxy server is password-protected specify:

```
objSoap.ConnectorProperty("ProxyUser") = "Proxy Username"
```

```
objSoap.ConnectorProperty("ProxyPassword") = "Proxy Password"
```

ColdFusion

3. Currently supported environments are ColdFusion 7 and ColdFusion 8. All necessary libraries are built into these environments.

Create a letter

Create a letter to send through the service. We've prepared a sample letter, or you can use a template to create your own:

- Download a sample letter in PDF format.
- Download a zipped file containing a Microsoft Word® 2007 template (.DOTX) and invoice sample file or a Microsoft Word® 2003 or older template (.DOT) and invoice sample file to create a letter of your own. Be sure not to move the recipient and return address text boxes.
- Download a zipped file containing an HTML template and an HTML sample file to create a letter of your own. In the sample file, you can see how to use DIVs in absolute position for the address blocks and how to force page breaks.
- Download a zipped file containing an OpenOffice Writer template to create a letter of your own. Be sure not to move the recipient and return address text boxes.

Run a code sample in your environment

1. Select an appropriate sample for your development environment to either *SendLetter* or *SendLetterAndAddress* on our Code Samples section.
2. Copy the sample to your development environment, replacing the variables with your details.
3. Adjust one of our sample letters to use for this test.
4. Run the code to send a letter.
5. If this has worked for you - congratulations! If not, see our Troubleshooting page or raise your issue on our Support Forum.

6. Review Your Activity Online

7. Login to your PostalMethods account at <https://cp.postalmethods.com>. Your letter should be visible on the *Overview* page (the page which you initially land on after you log in). You can click the little blue arrow on the left of each line to display more information.
8. The activity table also lets you view the PDF that was generated from your document and hide the transaction.