



# Qualys PCI

API User Guide  
Version 5.7

October 25, 2018

Copyright 2002-2018 by Qualys, Inc. All Rights Reserved.

Qualys and the Qualys logo are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.  
919 E Hillsdale Blvd  
4th Floor  
Foster City, CA 94404  
1 (650) 801 6100



# Table of Contents

## Preface

About Qualys .....	4
Qualys Support .....	4

## Chapter 1 - Welcome

Account Requirements .....	5
API Conventions .....	5

## Chapter 2 - New Merchant API

save_merchant.php Function .....	7
Parameters .....	7
Example .....	15
Generic Return XML .....	16
DTD for Generic Return .....	16

## Chapter 3 - Update Merchant API

update_merchant.php Function .....	18
Parameters .....	18
Example .....	20
Generic Return XML .....	20

## Chapter 4 - Get Merchant List API

get_merchant_list.php Function .....	21
Parameters .....	21
Example .....	21
Generic Return XML .....	22

## Appendix A - Country and State Codes

Country Codes .....	24
State Codes .....	25

# Preface

Using the Qualys API, third parties can integrate their own applications with Qualys cloud security and compliance solutions using an extensible XML interface. The API functions described in this guide are available to customers with Qualys PCI.

## About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated apps help businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications.

Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the [Cloud Security Alliance \(CSA\)](#). For more information, please visit [www.qualys.com](http://www.qualys.com)

## Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at [www.qualys.com/support/](http://www.qualys.com/support/)

# Chapter 1 - Welcome

The PCI compliance service from Qualys, Inc. provides an API for adding new merchants, allowing partners to automate the process of merchant enrollment. APIs are also provided for updating and listing merchant accounts.

The Qualys PCI APIs follow the same standards as other Qualys API functions. Before you begin, it's recommended that you review the account requirements and API conventions described in this chapter.

This chapter covers these topics:

[Account Requirements](#)

[API Conventions](#)

## Account Requirements

A Qualys PCI Admin account is required to add a merchant using the **save\_merchant.php** function. The application must authenticate using the administrator account credentials (username and password) as part of HTTP requests made to the Qualys PCI server.

If you need assistance with obtaining an administrator account, please contact Customer Support.

## API Conventions

Before you start using the Qualys PCI API, please review the following API conventions.

### PCI Server URL

The Qualys PCI server URL for API requests has this format:

```
https://pci.qualys.com/api/save_merchant.php?<parameters>
```

where <parameters> represents the user-specified input parameter name=value pairs for the new merchant request. The input parameters are described in detail in the next chapter.

### Authentication

The application must authenticate using Qualys PCI Admin account credentials (username and password) as part of the HTTP request. The credentials are transmitted using the "Basic Authentication Scheme" over HTTPS.

For more information, see the "Basic Authentication Scheme" section of RFC #2617:

```
http://www.faqs.org/rfcs/rfc2617.html
```

The exact method of implementing authentication will vary according to which programming language is used.

## **GET and POST Methods**

The Qualys PCI API allows users to submit input parameters (name=value pairs) using the GET and/or POST method. There are known limits for the amount of data that can be sent using the GET method, and these limits are dependent on the toolkit used.

## **Parameters in URLs**

API parameters, as documented in this user guide, should be specified one time for each URL. In the case where the same parameter is specified multiple times in a single URL, the last parameter takes effect and the previous instances are silently ignored.

## **URL Encoding in API Code**

You must URL encode variables when using the QualysPCI API. This is standard practice for HTTP communications. If your application passes special characters, like the single quote ('), parentheses, and symbols, they must be URL encoded.

For example, the pound (#) character cannot be used as an input parameter in URLs. If “#” is specified, the service returns an error. To specify the “#” character in a URL you must enter the encoded value “%23”. The “#” character is considered by browsers and other Internet tools as a separator between the URL and the results page, so whatever follows an un-encoded “#” character is not passed to the API server and returns an error.

## **UTF-8 Encoding**

The Qualys PCI API uses UTF-8 encoding.

## **URL Elements are Case Sensitive**

URL elements are case sensitive, including input parameter names and user-specified values.

## Chapter 2 - New Merchant API

As a Qualys PCI Partner, you can add new PCI merchant accounts using the API function called `save_merchant.php`. Once added, merchants receive accounts allowing them to log into the PCI compliance service using the PCI Merchant application. You can manage merchants using the Qualys PCI Admin web application. The PCI Compliance application allows you to search and view merchant accounts, edit account details, and reset account passwords. This chapter describes how to use the merchant API function. These topics are covered:

[save\\_merchant.php Function](#)

[Generic Return XML](#)

### save\_merchant.php Function

The `save_merchant.php` function is used to add a new merchant. Each new merchant is assigned an account type, either Trial or Customer. With a Trial account, the merchant can run scans and complete questionnaires, however PCI reports cannot be used for compliance certification. Your partner level determines whether you have privileges to create Customer type accounts. A Customer account is enabled with monthly billing (postpaid).

When making an API request, the application must authenticate using PCI Admin account credentials (user login name and password). The credentials are transmitted using the “Basic Authentication Scheme” over HTTPS.

Merchants are allowed to scan IP addresses that are included in the merchant account. When creating a new account, you have the option to add IP addresses when the merchant’s IP addresses are known. After account creation, merchants can add IP addresses to their own accounts up to the limit defined by the purchased IPs setting.

The GET or POST access method may be used to make a request using the `save_merchant.php`.

### Parameters

Several input parameters are required as indicated. This section describes the parameters, organized by merchant organization, merchant user, language, subscription and account manager.

#### Merchant Organization

These parameters specify merchant organization information.

Parameter	Description
<code>company={value}</code>	(Required) The merchant name. A maximum of 50 Unicode characters may be specified.
<code>addr1={value}</code>	(Required) The address line 1 for the merchant company address. A maximum of 50 Unicode characters may be specified.

Parameter	Description
addr2={value}	(Optional) The address line 2 for the merchant company address. A maximum of 50 Unicode characters may be specified.
country={value}	(Required) The merchant's country code. See "Country Codes" in Appendix A to find the appropriate country code.
city={value}	(Required) The merchant's city. A maximum of 50 Unicode characters may be specified.
state={value} (Optional)	<p>The merchant's state code. A valid value depends on the country code specified for the <b>country</b> parameter.</p> <p>You must enter a state code using the <b>state</b> parameter when the country code is one of: "United States of America", "Australia", "Canada" or "India". See "State Codes" in Appendix A to find the appropriate state code.</p> <p>For other country codes, a state code does not need to be specified with the <b>state</b> parameter. You may specify the state code "none" when a state code does not exist (and thus is not required).</p>
zipcode={value}	(Optional) The merchant's zip code. A maximum of 10 alphanumeric, Unicode characters may be specified.
comment_tag_1={value}	(Optional) A merchant comment tag which will be retrieved when you download merchant reports from the PCI Admin application. Note this value will not be exposed to the merchant in the PCI Merchant application. A maximum of 128 Unicode characters may be specified.
comment_tag_2={value}	(Optional) A merchant comment tag which will be retrieved when you download merchant reports from the PCI Admin application. Note this value will not be exposed to the merchant in the PCI Merchant application. A maximum of 128 Unicode characters may be specified.

## Merchant User

These parameters specify information about the primary user for the merchant account. Once the merchant account is created, the merchant may log into the PCI compliance service and add users to the subscription account.

Parameter	Description
prefix=Mr   Ms   Mrs	(Optional) The merchant user's courtesy title. One of these values can be entered: Mr, Ms, Mrs
firstname={value}	(Required) The merchant user's first name. A maximum of 50 Unicode characters may be specified.
lastname={value}	(Required) The merchant user's last name. A maximum of 50 Unicode characters may be specified.
title={value}	(Optional) The merchant user's title. A maximum of 50 Unicode characters may be specified.



Parameter	Description
phone={value}	(Optional) The merchant user's phone number. A maximum of 25 Unicode characters may be specified.
email={value}	(Required) The merchant user's email address in the format: <email_id>@domain. The service sends the new merchants account registration email to this email address. This value must be a properly formatted email address. A maximum of 128 ASCII characters may be specified.
username={value}	<p>(Optional) The merchant user's login ID. It's recommended that the user's email address is used. The login ID may consist of a maximum of 128 ASCII characters, and it must include the "@" character. For example: john.smith@qualys</p> <p>If not specified, the service creates a user name that matches the email address entered in the <b>email</b> parameter. For example, if the email address is rhall@qualys.com, then the service assigns this user name: rhall@qualys</p> <p>If not specified and the email address already exists in the application, the service creates a user name with the format: &lt;email_id&gt;@&lt;merchant_name&gt;, where &lt;email_id&gt; is the value entered in the <b>email</b> parameter, and &lt;merchant_name&gt; is the value entered in the <b>company</b> parameter.</p>

## Language

The language parameter is used to select a language for the merchant account. English is selected by default for a new account. The selected language is used for all merchant account users.

Parameter	Description
language={value}	<p>(Optional) The language setting for the merchant account. When unspecified, the language is set to English.</p> <p>A valid value is:</p> <ul style="list-style-type: none"> <li>en_US — for English, region United States (default)</li> <li>fr_CA — for French, region Canada</li> <li>fr_FR — for French, region France</li> <li>ja_JP — for Japanese, region Japan</li> <li>ko_KR — for Korean, region Korea</li> <li>pt_BR — for Portuguese, region Brazil</li> <li>ru_RU — for Russian, region Russia</li> </ul>

Service components are provided in the selected language as indicated below.

Language Setting	Description	Components
en_US (default)	English, region United States	In English: - user interface - vulnerability information - self-assessment questionnaires - online help
fr_CA	French, region Canada	In French (for region Canada): - user interface - self-assessment questionnaires  In English: - vulnerability information - online help
fr_FR	French, region France	In French (for region France): - user interface - self-assessment questionnaires - online help (PDF)  In English: - vulnerability information
ja_JP	Japanese, region Japan	In Japanese: - user interface - vulnerability information - self-assessment questionnaires - online help
ko_KR	Korean, region Korea	In Korean: - user interface - vulnerability information - self-assessment questionnaires - online help
pt_BR	Portuguese, region Brazil	In Portuguese (for region Brazil): - user interface - self-assessment questionnaires  In English: - vulnerability information - online help
ru_RU	Russian, region Russia	In Russian: - user interface  In English: - vulnerability information - self-assessment questionnaires - online help

Vulnerability information is visible in several places throughout the PCI Merchant application: the user's Current Vulnerabilities list, False Positives list, Scan Results Reports (PDF), Network Reports (PDF), and Self-Assessment Questionnaires.

Some reports are cached automatically in the language selected for the merchant: submitted Network Reports (PDF), submitted Questionnaires (PDF), and Web Application Scan Results (PDF).

An Admin user can change the language setting for a merchant account using the PCI Admin application. Remember the selected language affects all merchant account users. In the case where a merchant is defined for English and then an administrator changes the account to Japanese, for example, the merchant's cached reports (generated earlier when the account was set to English) remain in English and the merchant's new reports (generated after the account is set to Japanese) appear in Japanese.

### Merchant Level

The `merchant_level` parameter is used to specify the PCI merchant level for the merchant account, as defined by the PCI DSS. When unspecified, the merchant level is not defined for the merchant account.

Parameter	Description
<code>merchant_level={value}</code>	<p>(Optional) The merchant level as defined by the PCI DSS. When unspecified, a merchant level is not set for the merchant account. This parameter is required if you wish to enable the SAQ validation feature for the merchant.</p> <p>A valid value is an integer (1 to 4):</p> <ul style="list-style-type: none"><li>1 — for merchant level 1</li><li>2 — for merchant level 2</li><li>3 — for merchant level 3</li><li>4 — for merchant level 4</li></ul>

After account creation, the merchant level may be updated. A merchant has the ability to update the merchant level for their own merchant account using the PCI Merchant application (in account settings) or by an admin user using the PCI Admin application.

### Enable SAQ Validation

(Optional and only valid when you are a partner that has the SAQ validation feature enabled for your account.)

The SAQ validation feature is disabled for the merchant unless the `saq_validation_enabled=1` input parameter is specified.

When `saq_validation_enabled=1` is specified, this merchant will have the option to submit completed questionnaires to you for validation. For each completed questionnaire, the merchant may choose to either 1) submit the questionnaire directly to acquiring banks, or 2) submit the questionnaire to you for validation before they submit it to their acquiring banks.

Parameter	Description
<code>saq_validation_enabled={0 1}</code>	(Optional) Set this parameter to 1 to enable questionnaire validation for the merchant. When unspecified, questionnaire validation is not enabled for the merchant.  This parameter is not valid unless the <code>merchant_level</code> input parameter is also specified.

### Subscription Settings

These parameters specify subscription information for the merchant account.

Parameter	Description
<code>type=P   C</code>	(Required) The account type, either Prospect (Trial) or Customer. The value "P" specifies a Trial account. The value "C" specifies a Customer account that is enabled with monthly billing (postpaid). Your partner level and account settings determine whether you have privileges to create a Customer type account.  Some partners have privileges to create Trial accounts only. With a Trial account, a merchant is able to connect to the PCI Merchant application for a specified number of days. Reports generated using a Trial account cannot be used for certification with the PCI DSS.  Some partners have privileges to create Trial accounts and Customer accounts with monthly billing. With a Customer type account, a merchant is billed for their use of the PCI service on a monthly basis (postpaid).
<code>purchased_ips={value}</code>	(Optional) The number of purchased IP addresses. A valid value is an integer, which is greater than or equal to 0 and is less than 100 million. Do not enter commas. If 0 is specified, or the parameter is not specified at all, then the merchant cannot add IPs to their account (and thus cannot scan IPs).
<code>enddate={YYYY-MM-DD}</code>	(Optional) The account expiration date (some date in the future). This may be set in YYYY-MM-DD format. For example: 2007-09-12. If not specified, the service calculates the end date based on the account type.

Parameter	Description
ip_list={value}	<p>(Optional) A list of IP addresses and/or IP ranges. When no IPs are specified, the merchant user cannot perform network scans.</p> <p>For an IP range, separate the start and end IP with a hyphen (-). Separate multiple IPs and/or IP ranges with commas.</p> <p>Example IP range: 10.10.10.1-10.10.10.100</p> <p>Example IPs and IP ranges: 10.10.10.1-10.10.10.100,64.41.134.60, 64,41.134.66</p>

### Bank Information

These parameters specify bank information for the merchant account. The merchant has the option to add/edit this information using the PCI Merchant application.

Parameter	Description
bank_name_1={value}	<p>(Optional) The merchant's bank name.</p> <p>The bank name must match the name of a participating bank, as it is listed in the PCI Admin application. When provided, the merchant can auto submit completed questionnaires and network scan reports using the PCI Merchant application, and the bank views the submitted questionnaires and reports using the PCI Bank application.</p>
bank_ac_num_vis_mc_1={value}	<p>(Optional) The MasterCard or Visa "merchant account number" assigned to the merchant. A maximum of 32 characters (numeric) may be specified.</p>
bank_ac_num_amex_1={value}	<p>(Optional) The American Express "merchant account number" assigned to the merchant. A maximum of 32 characters (numeric) may be specified.</p>
bank_ac_num_discover_1={value}	<p>(Optional) The Discover "merchant account number" assigned to the merchant. A maximum of 32 characters (numeric) may be specified.</p>

## Questionnaire Access

These parameters specify questionnaire access for the merchant account.

Parameter	Description
quest_validation_type={value}	<p>(Optional) The questionnaire type access for the merchant account. This is associated with the PCI validation type. By default, a new merchant account allows access to all questionnaire types within the PCI Merchant application. When this parameter is unspecified, the merchant account will have full access to create and edit all questionnaires.</p> <p>To allow the merchant access to a certain questionnaire type only, specify the questionnaire type.</p> <p>A valid value is an integer (1 through 5, and 8):</p> <ul style="list-style-type: none"><li>1 — for SAQ A</li><li>2 — for SAQ B (Imprint Machines Only)</li><li>3 — for SAQ B (Stand-alone, Dial-out Terminals Only)</li><li>4 — for SAQ C</li><li>5 — for SAQ D (Merchant). When Type 5 is specified, only the SAQ D questionnaire for merchants will be available in the PCI Merchant application; the SAQ D questionnaire for service providers will not be available.</li><li>8 — for SAQ C-VT (Virtual Terminal)</li></ul>
quest_access_only={0 1}	<p>(Optional) A flag used to limit the merchant account access to the questionnaire only in the PCI Merchant application. By default, a new merchant account provides access to all functionality within the PCI Merchant application. When this parameter is unspecified, the merchant account will have full access to all functionality.</p> <p>To grant questionnaire access only to the merchant account, specify the value 1. This restricts merchant users access to questionnaire functionality; access to other functionality like launching scans and submitting network reports will not be permitted.</p> <p>When this parameter is set to 1, an error occurs if the parameter “purchased_ips” is specified with a value greater than 0.</p>

## Merchant's Account Manager

These parameters specify the Account Manager for the merchant account. The merchant account will be visible to the Account Manager in the PCI Admin application. When unspecified, an Account Manager is not set.

Parameter	Description
account_manager={value}	(Optional) Specifies the Admin ID of the administrator account that will manage the merchant account.  The <b>account_manager</b> and <b>account_manager_username</b> parameters are mutually exclusive and cannot be specified in the same request.
account_manager_username={value}	(Optional) Specifies the PCI Admin user's login ID for the administrator account that will manage the merchant account. For example: jdoe@qualys  The <b>account_manager</b> and <b>account_manager_username</b> parameters are mutually exclusive and cannot be specified in the same request.

## Example

A sample new merchant API request to add merchant "John Smith" is shown below. This request was made by PCI Admin user "erin@qualys", as indicated in the XML output.

```
https://pci.qualys.com/api/save_merchant.php?company=Acme%20Stars&
addr1=Hillsdale%20Blvd&addr2=4th%20Floor&city=Foster%20City&state=
CA&country=United%20States%20of%20America&zipcode=94404&type=C&pre
fix=Mr&firstname=John&lastname=Smith&title=CFO&phone=16508006121&u
sername=jsmith@acme&purchased_ips=10&email=jsmith@acme.com&ip_list
=10.10.10.1&bank_name_1=Worldwide%20Bank&bank_ac_num_vis_mc_1=1231
23123123123123123123&comment_tag_1=domain%20is%20corp1.us.com
```

The XML output uses the DTD "generic\_return.dtd". Sample XML output is below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE GENERIC_RETURN SYSTEM "/static/dtd/generic_return.dtd">
<GENERIC_RETURN>
<API name="save_merchant.php" username="erin@qualys" at="2018-10-
02T19:28:46Z"/>
<RETURN status="SUCCESS" number="" account_id="3442"
account_username="jsmith@acme">User jsmith@acme has been
successfully created.</RETURN>
</GENERIC_RETURN>
```

## Generic Return XML

The generic return is XML output returned from the `save_merchant.php` request. The DTD “`simple_return.dtd`” can be found at the following URL:

[https://pci.qualys.com/static/dtd/generic\\_return.dtd](https://pci.qualys.com/static/dtd/generic_return.dtd)

The DTD output and XPaths are described below.

### DTD for Generic Return

A recent DTD for the generic return output (`generic_return.dtd`) is shown below.

```
<!-- QUALYS GENERIC_RETURN DTD -->
<!ELEMENT GENERIC_RETURN (API,RETURN)>

<!-- "name" is the name of API -->
<!-- "at" attribute is the current platform date and time -->
<!ELEMENT API (#PCDATA)>
<!ATTLIST API
    name CDATA #REQUIRED
    username CDATA #REQUIRED
    at CDATA #REQUIRED>

<!-- the PCDATA contains an explanation of the status -->
<!ELEMENT RETURN (#PCDATA)*>
<!ATTLIST RETURN
    status (FAILED|SUCCESS|WARNING) #REQUIRED
    number CDATA #IMPLIED
    account_id CDATA #IMPLIED
    account_username CDATA #IMPLIED>

<!-- EOF -->
```

### XPaths for Generic Return

This section describes the XPaths for the generic return output (`generic_return.dtd`).

XPath	element specifications / notes
<code>/GENERIC_RETURN</code>	(API, RETURN)
<code>/GENERIC_RETURN/API</code>	(#PCDATA)
<code>name</code>	(Required) The API function name, in this case “ <code>save_merchant.php</code> ”.
<code>username</code>	(Required) The user name of the administrator account that made the API request.
<code>at</code>	(Required) The date and time of the API request in this format: <code>YYYY-MM-DDThh:mm:ssZ</code>



<b>XPath</b>	<b>element specifications / notes</b>
/GENERIC_RETURN/RETURN	(#PCDATA) A message that describes the API request status. Upon success, the message is: "User <merchant user login name> has been successfully created."
status	(Required) The status of the API request. The return value can be: SUCCESS, FAILED OR WARNING.
number	(Implied) The error code, if applicable. This attribute is always empty when the status is SUCCESS.
account_id	(Implied) The account ID is a number that uniquely identifies the subscription in the PCI application.
account_username	(Implied) The merchant user login name for the new merchant account created by the API request.

## Chapter 3 - Update Merchant API

You can make certain updates to a merchant account using the API function called `update_merchant.php`. For example, you can use this function for account renewal by extending a merchant account's expiration date, adding more purchased IP addresses or web applications, changing the account status, and changing an account from Prospect to Customer.

This chapter describes how to update merchant account data. These topics are covered:

[update\\_merchant.php Function](#)

[Generic Return XML](#)

### update\_merchant.php Function

The `update_merchant.php` function is used to update certain merchant account settings, including expiration date, total number of IP addresses or web applications purchased, account type (Prospect or Customer), account manager, and enabling/disabling the Web Application Scanning (WAS) module.

When making an API request, the application must authenticate using PCI Admin account credentials (user login name and password). The credentials are transmitted using the "Basic Authentication Scheme" over HTTPS.

The GET or POST access method may be used to make a request using the `update_merchant.php`.

### Parameters

These parameters specify merchant account details to update.

Parameters	Description
<code>merchant_id={value}</code>	(Required) Specifies the merchant ID of the merchant account you wish to update.
<code>enable_was=true false</code>	(Optional) Specifies whether to enable or disable the Web Application Scanning (WAS) module for the merchant. Specify "true" to enable the WAS module. Specify "false" to disable the WAS module.
<code>expiration_date={value}</code>	(Optional) Specifies the date the merchant account is set to expire (a date in the future). The date is specified in this format: YYYY-MM-DD.  If the merchant account being updated is currently expired, then the account status will change to "active".
<code>was_expiration_date={value}</code>	(Optional) Specifies the date the WAS module trial period is set to expire (a date in the future that is no later than the account expiration date). The date is specified in this format: YYYYMM-DD.

Parameters	Description
status={value}	(Optional) Specifies a change to the merchant account status. Specify “deactivate” to make the account inactive. Deactivated users cannot log in to the PCI module. Specify “activate” to make an inactive account active again.
comments={value}	(Required) Specifies comments to be saved with the merchant account. Comments are useful for account management and tracking actions taken. The Comment History section in the PCI Admin application shows previously saved comments. Each comment entry identifies the admin user who added the comment and the date when the comment was made.
total_ips={value}	(Optional) Specifies the total number of IP addresses the merchant has purchased to scan. For example, if you want to add 10 IPs to the account to bring the new total to 50 IPs, then specify total_ips=50.
total_web_apps={value}	(Optional) Specifies the total number of web applications the merchant has purchased to scan. For example, if you want to add 3 web applications to the account to bring the new total to 7 web applications, then specify total_web_apps=7.
account_type=c   p	(Optional) The account type, either Customer or Prospect (Trial). The value “c” specifies a Customer account. The value “p” specifies a Prospect account.
account_manager={value}	(Optional) Specifies the Admin ID of the administrator account that will manage the merchant account.  The <b>account_manager</b> and <b>account_manager_username</b> parameters are mutually exclusive and cannot be specified in the same request.
account_manager_username={value}	(Optional) Specifies the PCI Admin user’s login ID for the administrator account that will manage the merchant’s account. For example: jdoe@qualys  The <b>account_manager</b> and <b>account_manager_username</b> parameters are mutually exclusive and cannot be specified in the same request.

## Example

A sample update merchant API request to update the merchant with merchant ID “574” to change the expiration date to the end of 2018 is shown below. This request was made by PCI Admin user “erin@qualys”, as indicated in the sample XML output.

```
https://pci.qualys.com/api/update_merchant.php?merchant_id=574&exp  
iration_date=2018-12-31&comments=extending+expiration+date+to+2018
```

A sample update merchant API request to update the merchant with merchant ID “574” to deactivate the account is shown below. This request was made by PCI Admin user “erin@qualys”, as indicated in the sample XML output.

```
https://pci.qualys.com/api/update_merchant.php?merchant_id=574&sta  
tus=deactivate&comments=deactivate+user+temporarily
```

## Generic Return XML

Sample “Success” XML output is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>  
<GENERIC_RETURN>  
<API name="update_merchant.php" username="erin@qualys" at="2017-  
12-03T03:34:42Z" />  
<RETURN status="SUCCESS" number="" merchant_id="574">Merchant 574  
has been successfully updated.</RETURN>  
</GENERIC_RETURN>
```

Sample “Failed” XML output is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>  
<GENERIC_RETURN>  
<API name="update_merchant.php" username="erin@qualys" at="2017-  
12-03T03:35:20Z" />  
<RETURN status="FAILED" number="">Account Type should be  
C/P</RETURN>  
</GENERIC_RETURN>
```

## Chapter 4 - Get Merchant List API

You can get a list of your merchant accounts with detailed account information using the API function called `get_merchant_list.php`.

This chapter describes how to list merchant accounts. These topics are covered:

[get\\_merchant\\_list.php Function](#)

[Generic Return XML](#)

### get\_merchant\_list.php Function

The `get_merchant_list.php` function is used to return a list of your merchants with detailed account information. You can make a request to return all merchant accounts (under your PCI partner name) or specify a merchant ID to only return details for a specific merchant account.

When making an API request, the application must authenticate using PCI Admin account credentials (user login name and password). The credentials are transmitted using the “Basic Authentication Scheme” over HTTPS.

The GET or POST access method may be used to make a request using the `get_merchant_list.php`.

### Parameters

These parameters may be specified to filter the list of merchants returned in the XML output.

Parameters	Description
<code>billing_type=M   Y</code>	(Optional) Show only merchant accounts with a certain billing type: Monthly or Yearly. The value “M” specifies Monthly billing. The value “Y” specifies Yearly billing.
<code>merchant_id={value}</code>	(Optional) Show only the merchant account with a merchant ID equal to the merchant ID specified.
<code>last_update_date={value}</code>	(Optional) Show only merchant accounts updated since a certain date. The date is specified in this format: YYYY-MM-DD.
<code>qualys_merchants_only=true</code>	(Optional) For Qualys internal use only.

### Example

A sample get merchant list API request to list all merchant accounts that have yearly billing and have been updated since January 1, 2017 is shown below. This request was made by PCI Admin user “erin@qualys”, as indicated in the sample XML output.

```
https://pci.qualys.com/api/get_merchant_list.php?billing_type=y&last_update_date=2017-01-01
```

## Generic Return XML

Sample "Success" XML output is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<GENERIC_RETURN>
  <API name="get_merchant_list.php" username="erin@qualys"
at="2017-12-03T03:21:45Z" />
  <MERCHANTS>
    <MERCHANT>
      <MERCHANT_NAME><![CDATA[Acme, Inc.]]></MERCHANT_NAME>
      <CONTACT_NAME><![CDATA[Joe Smith]]></CONTACT_NAME>
      <CONTACT_EMAIL><![CDATA[jsmith@acme.com]]></CONTACT_EMAIL>
      <STATE>California</STATE>
      <COUNTRY>United States of America</COUNTRY>
      <STATUS>Active</STATUS>
      <STATUS_CHANGE_DATE>2017-08-11 01:16:21</STATUS_CHANGE_DATE>
      <TYPE>Customer</TYPE>
      <TYPE_CHANGE_DATE>2017-12-03 01:01:37</TYPE_CHANGE_DATE>
      <BILLING_TYPE>Y</BILLING_TYPE>
      <NUM_IPS_PURCHASED>300</NUM_IPS_PURCHASED>
      <NUM_IPS_ADDED>45</NUM_IPS_ADDED>
      <COMMENT_TAG_1></COMMENT_TAG_1>
      <COMMENT_TAG_2></COMMENT_TAG_2>
      <CREATION_DATE>2016-10-10 18:28:03</CREATION_DATE>
      <EXPIRATION_DATE>2018-12-10 08:00:00</EXPIRATION_DATE>
      <SCAN_COMPLIANCE_STATUS>Not
compliant</SCAN_COMPLIANCE_STATUS>
      <QUESTION_COMPLIANCE_STATUS>Compliant</QUESTION_COMPLIANCE_STATUS>
      <SUA_DATE>2017-10-10 18:36:39</SUA_DATE>
      <LAST_SCAN_SUBMIT_DATE>2017-09-03
23:11:32</LAST_SCAN_SUBMIT_DATE>
      <NEXT_SCAN_COMPLIANCE_DATE>2017-12-02
23:11:32</NEXT_SCAN_COMPLIANCE_DATE>
      <LAST_QUESTION_SUBMIT_DATE>2017-04-14
17:56:16</LAST_QUESTION_SUBMIT_DATE>
      <NEXT_QUESTION_COMPLIANCE_DATE>2018-04-14
17:56:16</NEXT_QUESTION_COMPLIANCE_DATE>
      <BANK_NAME></BANK_NAME>
      <PARTNER_NAME><![CDATA[Qualys, Inc]]></PARTNER_NAME>
      <PARTNER_ID>100</PARTNER_ID>
      <WAS_ENABLED>YES</WAS_ENABLED>
      <WAS_APP_PURCHASED>12</WAS_APP_PURCHASED>
      <WAS_APP_ADDED>6</WAS_APP_ADDED>
      <WAS_TRIAL>NO</WAS_TRIAL>
      <WAS_TRIAL_EXP_DATE>2018-01-01 08:00:00</WAS_TRIAL_EXP_DATE>
      <QUESTIONNAIRE_ACCESS_ONLY>NO</QUESTIONNAIRE_ACCESS_ONLY>
```

```
<MERCHANT_ID>574</MERCHANT_ID>
<LAST_NETWORK_SCAN_DATE></LAST_NETWORK_SCAN_DATE>
<LAST_WAS_SCAN_DATE></LAST_WAS_SCAN_DATE>
<ADDRESS1><![CDATA[123 Main Street]]></ADDRESS1>
<ADDRESS2><![CDATA[]]></ADDRESS2>
<CITY>Anywhere</CITY>
<ZIP_CODE>12345</ZIP_CODE>
<CONTACT_PHONE>111 222 3333</CONTACT_PHONE>
<CONTACT_LOGIN><![CDATA[jsmith@acme]]></CONTACT_LOGIN>
<QWEB_SUBSCRIPTION_ID>100</QWEB_SUBSCRIPTION_ID>
</MERCHANT>
</MERCHANTS>
</GENERIC_RETURN>
```

Sample “Failed” XML output is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <GENERIC_RETURN>
    <API name="get_merchant_list.php" username="erin@qualys"
at="2017-12-03T03:22:27Z" />
    <RETURN status="FAILED" number="">Invalid MerchantId
Provided.</RETURN>
  </GENERIC_RETURN>
```

## Appendix A - Country and State Codes

This appendix describe the country and state codes that may be specified when defining a new merchant using the `save_merchant.php` function. For information on using the `save_merchant.php` function, see [Chapter 2 - New Merchant API](#) earlier in this document.

Valid country and state codes are described below.

### Country Codes

Valid country codes:

Afghanistan | Albania | Algeria | Andorra | Angola | Anguilla | Antarctica | Antigua and Barbuda | Argentina | Armenia | Aruba | Australia | Austria | Azerbaijan | Bahamas | Bahrain | Bangladesh | Barbados | Belarus | Belgium | Belize | Benin | Bermuda | Bhutan | Bolivia | Bosnia-Herzegovina | Botswana | Bouvet Island | Brazil | British Indian Ocean Territory | Brunei Darussalam | Bulgaria | Burkina Faso | Burundi | Cambodia | Cameroon | Canada | Cape Verde | Cayman Islands | Central African Republic | Chad | Chile | China | Christmas Island | Cocos (Keeling) Islands | Colombia | Comoros | Congo | Cook Islands | Costa Rica | Cote D'Ivoire | Croatia | Cuba | Cyprus | Czech Republic | Denmark | Djibouti | Dominica | Dominican Republic | East Timor | Ecuador | Egypt | El Salvador | Equatorial Guinea | Estonia | Ethiopia | Faeroe Islands | Falkland Islands (Malvinas) | Fiji | Finland | France | French Guiana | French Polynesia | French Southern Territories | Gabon | Gambia | Georgia | Germany | Ghana | Gibraltar | Greece | Greenland | Grenada | Guadeloupe | Guatemala | Guernsey, C.I. | Guinea | Guinea-Bissau | Guyana | Haiti | Heard and McDonald Islands | Honduras | Hong Kong | Hungary | Iceland | India | Indonesia | Iran (Islamic Republic of) | Iraq | Ireland | Isle of Man | Israel | Italy | Jamaica | Japan | Jersey, C.I. | Jordan | Kazakhstan | Kenya | Kiribati | Korea | Kuwait | Kyrgyzstan | Lao Peoples Democratic Republic | Latvia | Lebanon | Lesotho | Liberia | Libyan Arab Jamahiriya | Liechtenstein | Lithuania | Luxembourg | Macau | Macedonia | Madagascar | Malawi | Malaysia | Maldives | Mali | Malta | Marshall Islands | Martinique | Mauritania | Mauritius | Mexico | Micronesia, Fed. States of | Moldova, Republic of | Monaco | Mongolia | Montserrat | Morocco | Mozambique | Myanmar | Namibia | Nauru | Nepal | Netherland Antilles | Netherlands | Neutral Zone (Saudi/Iraq) | New Caledonia | New Zealand | Nicaragua | Niger | Nigeria | Niue | Norfolk Island | Northern Mariana Islands | Norway | Oman | Pakistan | Palau | Panama Canal Zone | Panama | Papua New Guinea | Paraguay | Peru | Philippines | Pitcairn | Poland | Portugal | Puerto Rico | Qatar | Reunion | Romania | Russia | Rwanda | Saint Kitts and Nevis | Saint Lucia | Samoa | San Marino | Sao Tome and Principe | Saudi Arabia | Senegal | Seychelles | Sierra Leone | Singapore | Slovak Republic | Slovenia | Solomon Islands | Somalia | South Africa | Spain | Sri Lanka | St. Helena | St. Pierre and Miquelon | St. Vincent and the Grenadines | Sudan | Suriname | Svalbard and Jan Mayen Islands | Swaziland | Sweden | Switzerland | Syrian Arab Republic | Taiwan | Tajikistan | Tanzania, United Republic of | Thailand | Togo | Tokelau | Tonga | Trinidad and Tobago | Tunisia | Turkey | Turkmenistan | Turks and Caicos Islands | Tuvalu | U.S. Minor Outlying Islands | Uganda | Ukraine | United Arab Emirates | United Kingdom | United States of America | Uruguay | Uzbekistan | Vanuatu | Vatican City State | Venezuela | Vietnam | Virgin Islands (British) | Wallis and Futuna Islands | Western Sahara | Yemen | Yugoslavia | Zaire | Zambia | Zimbabwe



## State Codes

### State Codes for United States

Value state codes when country is “United States of America”:

No State | Alabama | Alaska | Arizona | Arkansas | Armed Forces Asia | Armed Forces Europe | Armed Forces Pacific | California | Colorado | Connecticut | Delaware | District of Columbia | Florida | Georgia | Hawaii | Idaho | Illinois | Indiana | Iowa | Kansas | Kentucky | Louisiana | Maine | Maryland | Massachusetts | Michigan | Minnesota | Mississippi | Missouri | Montana | Nebraska | Nevada | New Hampshire | New Jersey | New Mexico | New York | North Carolina | North Dakota | Ohio | Oklahoma | Oregon | Pennsylvania | Rhode Island | South Carolina | South Dakota | Tennessee | Texas | Utah | Vermont | Virginia | Washington | West Virginia | Wisconsin | Wyoming

### State Codes for Australia

Valid state codes when country is “Australia”:

No State | New South Whales | Northern Territory | Queensland | Tasmania | Victoria | Western Australia

### State Codes for Canada

Valid state codes when country is “Canada”:

No State | Alberta | British Columbia | Manitoba | New Brunswick | Newfoundland | Northwest Territories | Nova Scotia | Nunavut | Ontario | Prince Edward Island | Quebec | Saskatchewan | Yukon

### State Codes for India

Valid state codes when country is “India”:

No State | Andhra Pradesh | Andaman and Nicobar Islands | Arunachal Pradesh | Assam | Bihar | Chandigarh | Chattisgarh | Dadra and Nagar Haveli | Daman and Diu | Delhi | Goa | Gujarat | Haryana | Himachal Pradesh | Jammu and Kashmir | Jharkhand | Karnataka | Kerala | Lakshadweep | Madhya Pradesh | Maharashtra | Manipur | Meghalaya | Mizoram | Nagaland | Orissa | Pondicherry | Punjab | Rajasthan | Sikkim | Tamil Nadu | Tripura | Uttar Pradesh | Uttaranchal | West Bengal