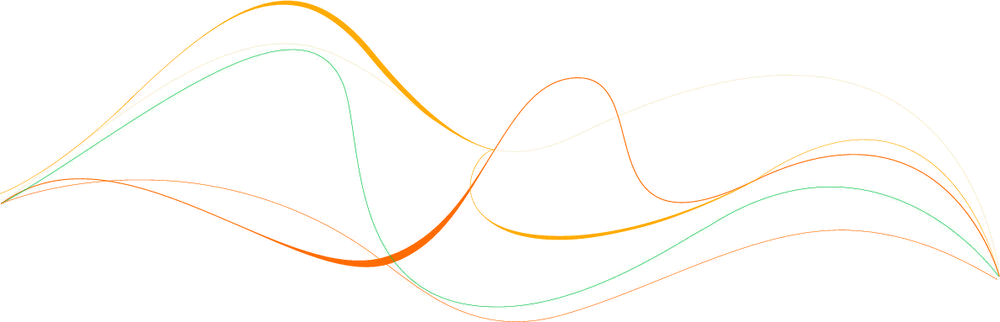
Reporting & Admin API

Integration Guideline

Protocol Version 1.02

Document Version 0.5



# Contents

[Contents 2](#_Toc371400541)

[1 Introduction 4](#_Toc371400542)

[1.1 How to use this integration guide 4](#_Toc371400543)

[1.2 API Request and Response Formats 4](#_Toc371400544)

[1.3 When no XML field is supplied or <command> element is “version” 5](#_Toc371400545)

[2 The different <command> options 6](#_Toc371400546)

[2.1 Admin Command - version 8](#_Toc371400547)

[2.2 Admin Command - createUser 9](#_Toc371400548)

[2.3 Admin Command - deleteUser 11](#_Toc371400549)

[2.4 Admin Command - lockUser 12](#_Toc371400550)

[2.5 Admin Command - logAccountActivity 13](#_Toc371400551)

[2.6 Admin Command - logoutUser 14](#_Toc371400552)

[2.7 Admin Command - unlockUser 15](#_Toc371400553)

[2.8 Admin Command - setUserPassword 16](#_Toc371400554)

[2.9 Admin Command - setUserRights 17](#_Toc371400555)

[2.10 Admin Command - set3DSecureStatus 19](#_Toc371400556)

[2.11 Admin Command - setAVSCV2Status 20](#_Toc371400557)

[2.12 Admin Command - set3DSecureRules 21](#_Toc371400558)

[2.13 Admin Command - setAVSCV2Rules 22](#_Toc371400559)

[2.14 Admin Command - setExpiryDate 23](#_Toc371400560)

[2.15 Admin Command - setVendorDetails 25](#_Toc371400561)

[2.16 Admin Command – setUserDetails 27](#_Toc371400562)

[2.17 Admin Command - addBlockedCardRanges 28](#_Toc371400563)

[2.18 Admin Command - addBlockedCountries 29](#_Toc371400564)

[2.19 Admin Command - addBlockedIPs 30](#_Toc371400565)

[2.20 Admin Command - addBlockedIssuingCountries 31](#_Toc371400566)

[2.21 Admin Command - addValidIPs 32](#_Toc371400567)

[2.22 Admin Command - addRestrictedDomains 33](#_Toc371400568)

[2.23 Admin Command - setVendorSurcharges 34](#_Toc371400569)

[2.24 Admin Command - deleteAVSCV2Rules 36](#_Toc371400570)

[2.25 Admin Command - delete3DSecureRules 37](#_Toc371400571)

[2.26 Admin Command - deleteBlockedCardRanges 38](#_Toc371400572)

[2.27 Admin Command - deleteBlockedCountries 39](#_Toc371400573)

[2.28 Admin Command - deleteBlockedIPs 40](#_Toc371400574)

[2.29 Admin Command - deleteBlockedIssuingCountries 41](#_Toc371400575)

[2.30 Admin Command - deleteValidIPs 42](#_Toc371400576)

[2.31 Admin Command - setPaymentPageTemplates 43](#_Toc371400577)

[2.32 Admin Command - setPaypalAccount 44](#_Toc371400578)

[2.33 Admin Command - deletePaypalAccount 45](#_Toc371400579)

[2.34 Admin Command - deleteRestrictedDomain 46](#_Toc371400580)

[2.35 Admin Command - deleteVendorSurcharges 46](#_Toc371400581)

[2.36 Reporting Command - doesUserExist 48](#_Toc371400582)

[2.37 Reporting Command - getUserRights 49](#_Toc371400583)

[2.38 Reporting Command - getAccountLogs 51](#_Toc371400584)

[2.39 Reporting Command - getVendorUserList 52](#_Toc371400585)

[2.40 Reporting Command - getVendorDetails 54](#_Toc371400586)

[2.41 Reporting Command - getVendorPProDetail 55](#_Toc371400587)

[2.42 Reporting Command - getVendorPaymentSystems 56](#_Toc371400588)

[2.43 Reporting Command - getValidIPs 58](#_Toc371400589)

[2.44 Reporting Command - getURLs 59](#_Toc371400590)

[2.45 Reporting Command - get3DSecureStatus 61](#_Toc371400591)

[2.46 Reporting Command - get3DSecureRules 62](#_Toc371400592)

[2.47 Reporting Command - getAVSCV2Status 63](#_Toc371400593)

[2.48 Reporting Command - getAVSCV2Rules 64](#_Toc371400594)

[2.49 Reporting Command - getRestrictedDomains 65](#_Toc371400595)

[2.50 Reporting Command - getBatchDetail 66](#_Toc371400596)

[2.51 Reporting Command - getBatchList 71](#_Toc371400597)

[2.52 Reporting Command - getCardType 73](#_Toc371400598)

[2.53 Reporting Command - getCardDetails 74](#_Toc371400599)

[2.54 Reporting Command – getCurrencies 76](#_Toc371400600)

[2.55 Reporting Command - getMerchantAccounts 77](#_Toc371400601)

[2.56 Reporting Command - getTransactionCardDetails 79](#_Toc371400602)

[2.57 Reporting Command - getTransactionDetail 80](#_Toc371400603)

[2.58 Reporting Command - getTransactionIPDetails 87](#_Toc371400604)

[2.59 Reporting Command - getTransactionList 88](#_Toc371400605)

[2.60 Reporting Command - getTransactionSummary 94](#_Toc371400606)

[2.61 Reporting Command - getTransactionTypes 97](#_Toc371400607)

[2.62 Reporting Command - getRelatedTransactions 98](#_Toc371400608)

[2.63 Reporting Command - getGiftAidReport 102](#_Toc371400609)

[2.64 Reporting Command - getAllCountries 104](#_Toc371400610)

[2.65 Reporting Command - getBlockedCardRanges 105](#_Toc371400611)

[2.66 Reporting Command - getBlockedCountries 106](#_Toc371400612)

[2.67 Reporting Command - getBlockedIPs 107](#_Toc371400613)

[2.68 Reporting Command - getBlockedIssuingCountries 108](#_Toc371400614)

[2.69 Reporting Command - getTokenCount 109](#_Toc371400615)

[2.70 Reporting Command - getTokenDetails 110](#_Toc371400616)

[2.71 Reporting Command - getT3MDetail 112](#_Toc371400617)

[2.72 Reporting Command - getUserDetails 114](#_Toc371400618)

[2.73 Reporting Command - getFraudScreenDetail 115](#_Toc371400619)

[2.74 Reporting Command - getVendorSurcharges 116](#_Toc371400620)

[3 Using Special Characters 117](#_Toc371400621)

[4 Full <errorcode> list 117](#_Toc371400622)

# Introduction

This document details the full set of requests and responses available from the Reporting & Admin API system. This API allows your back-office systems to interact with the Sage Pay payment systems in the same manner as a person with a web browser might use our My Sage Pay administration area.

The API uses the same user accounts and privilege system as My Sage Pay, and will allow account administration and reporting functions, but with a finer degree of granularity than the My Sage Pay screens, with all messages in real-time in a consistent XML format.

All requests are sent to one gateway at the following address:

Test Server: <https://test.sagepay.com/access/access.htm>

Live Server: <https://live.sagepay.com/access/access.htm>

## How to use this integration guide

All the various commands listed within this guide are outlined in the same structure. You will find a brief overview of what the command is for along with both the request and response protocol when using the command. Along with the protocol, you will find some example code for the request and response to help you.

Please be aware of the following symbols throughout this documentation:

 The user name provided must have administrative privileges to run this command.

 This command doesn’t require the user to have administrator rights.

## API Request and Response Formats

The API requests are HTTPS POSTs to the URLs above, containing a single FORM field called **XML**. It is important to note that these are not requests in XML format, but regular HTTPS POSTs with the content type “application/x-www-form-urlencoded” with the XML requests detailed below in a FORM field called XML

The XML field will contain the XML message, which always takes the following format:

<vspaccess>­­

    <command>command name</command>

    <vendor>vendor name</vendor>

    <user>user name</user>

    <other command specific parameters in here..../>

    <signature>MD5 Hash Signature</signature>

</vspaccess>

The <signature> value is created by taking the rest of the contents of the <vspaccess> node, appending an element called <password> which contains the user account's password, and MD5 hashing the message.  So in the above example:

MD5 Hash Signature =

MD5Hash(

    "<command>whatever</command>

    <vendor>vendor name</vendor>

    <user>user name</user>

    <other command specific params in here..../>

    <password>user password</password>")

The response format will always be a XML v1.0 ISO-8859-1 encoded document with a root node called <vspaccess>. i.e.:

<?xml version="1.0" encoding="ISO-8859-1" ?>

<vspaccess>

    <errorcode>nnnn</errorcode>

    <error>Text error message if errorocode is not 0000</error>

    <other command specific results here..../>

<timestamp>DD/MM/YYYY HH:MM:SS</timestamp>

</vspaccess>

The <errorcode> field is always present, and for successful requests will always be **0000**. All other values indicate an error, and the <error> element will provide a human readable interpretation of the error code. A list of error codes is given in Appendix A below.

The <timestamp> will also always be present and contains the time that the response was sent back to the requesting process.

## When no XML field is supplied or <command> element is “version”

If no XML field is supplied in the POST, or if the <command> element is set to “version” (i.e. <command>version</command>) the system will simply return the version number of the active Reporting & Admin API system, and the timestamp. For example:

<?xml version="1.0" encoding="ISO-8859-1" ?>

<vspaccess>

<errorcode>0000</errorcode>

<version>1.02</version>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

# The different <command> options

This section lists the commands that can be passed in the <command> element of the request. This list will grow with subsequent releases, so please ensure you are working from the most recent release of this document.

Each section lists the command and the required and optional parameters that form the complete request in the table below it. This table should be mapped to valid API XML request by appending the <signature> element and <vspaccess> node as described in section 1.1 above. The response from the API, again with optional components, is listed immediately after.

For example, the **getURLs** command has the following request format:

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getURLs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <system> | String  (10 characters) | The system for which the URLs are being requested. Must be either **VSPServer**, **VSPDirect** or **VSPForm** |
| <txtypes> (optional) | Node | Container for the list of transactions types for which the URLs are being requested. If not provided, all service URLs are returned. |
| └<txtype>  (optional) | String  (20 characters) | List of transaction types for which the URLs are being requested. |

This would translate to an XML request like this:

<vspaccess>

<command>getURLs</command>

<vendor>vendorname</vendor>

<user>username</user>

<system>VSPDirect</system>

<txtypes>

<txtype>payment</txtype>

<txtype>refund</txtype>

</txtypes>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

This would request the URLs for the payment and refund services for the Direct payment system.

The Response to that request is summarised in a table like this:

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <urls> | Node | Container for the URLs being returns |
| └<url> | Node | Container for a single URL with the service it matches |
| └<txtype> | String  (20 characters) | The name of the transaction type for which this URL maps |
| └<serviceURL> | String - FQDN  (255 characters) | Fully Qualified Domain Name URL to the service named above. |

This would map to the following response in valid XML:

<?xml version="1.0" encoding="ISO-8859-1" ?>

<vspaccess>

<errorcode>0000</errorcode>

<urls>

<url>

<txtype>payment</txtype>

<serviceURL>https://test.sagepay.com/payment</serviceURL>

</url>

<url>

<txtype>refund</txtype>

<serviceURL>https://test.sagepay.com/refund</serviceURL>

</url>

</urls>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - version

This command returns the version number of the Reporting & Admin API system to which the message is sent. This command is executed by default if no XML parameter is specified in the request POST.

 This command doesn’t require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  version | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <version> | Alphanumeric  (10 characters) | The current version number of the Reporting & Admin API system being accessed. |

**Sample XML Input**

<vspaccess>

<command>version</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<version>1.02</version>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - createUser

The createUser command allows the creation of a user account accessible to both My Sage Pay and this API. It mimics the user account creation process in the Settings section in My Sage Pay.

 The user name provided must have administrative privileges to run this command.

See also: deleteUser, setUserRights, lockUser, unlockUser, logoutUser, setUserPassword

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  createUser | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <newuser> | Node | Container for the new user account parameters. |
| └<username> | Alphanumeric  (20 characters) | The new user’s user name. Can only contain letters and numbers. |
| └<password> | String  (32 characters) | The new user’s password. CASE SENSITIVE and can contain ASCII characters |
| └<accessrights> | Node | Container for the access rights |
| └<access> | String  (specific values) | Access values can be:  **Transactions** – Access to the transaction area  **Reports** – Access to the reporting area  **Terminal** – Can process Terminal transactions  **Admin** – USER HAS ADMIN RIGHTS!  **CanViewAll** – Can view other user’s transactions as well as their own.  **CanRefund**, **CanRelease**, **CanAbort**, **CanRepeat**, **CanVoid** and **CanManual** – user can perform these transaction types if passed |
| └<homepage>  (optional) | String  (20 characters) | The initial page the user is presented with when they log in. The following values can be used:  **transactions.msp** – the report generation page  **settings.msp** – the account administration page  **terminal.msp** – the virtual terminal page  **search.msp** – search transactions |
| └<emailaddress>  (optional) | String  (255 characters) | An email address associated with the User account. Each user must have a unique email address. |
| └<firstname>  (optional) | String  (70 characters) | The user’s first name. |
| └<lastname>  (optional) | String  (70 characters) | The user’s last name. |
| └<commsoptin>  (optional) | String  (3 characters) | Determines whether a user wants to opt in or out of receiving communications from Sage Pay. The field should contain either **YES** or **NO**. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>createUser</command>

<vendor>vendorname</vendor>

<user>username</user>

<newuser>

<username>username2</username>

<password>mypassword</password>

<accessrights>

<access>TRANSACTIONS</access>

<access>REPORTS</access>

</accessrights>

<homepage>settings.msp</homepage>

<emailaddress>test@test.com</emailaddress>

**<**firstname>first</firstname>

<lastname>last</lastname>

<commsoptin>YES</commsoptin>

</newuser>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteUser

The deleteUser command allows the removal of a user from both My Sage Pay and this API. It mimics the user deletion process in the Settings section of the My Sage Pay screens.

 The user name provided must have administrative privileges to run this command.

See also: createUser, setUserRights, lockUser, unlockUser, logoutUser, setUserPassword

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteUser | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user name of the user to be deleted. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteUser</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - lockUser

This command will lock out a specified user account until an administrator unlocks it. It will also log out the specific user if they are active at the time this command is issued.

 The user name provided must have administrative privileges to run this command.

See also: createUser, deleteUser, setUserRights, unlockUser, logoutUser, setUserPassword

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  lockUser | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user account to lock out. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>lockUser</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - logAccountActivity

This command creates in the account logs for the particular account. These logs are visible within My Sage Pay.

 The user name provided must have administrative privileges to run this command.

See also: getAccountLogs

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  logAccountActivity | The service being requested. |
| <vendor> | String  (16 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | String  (20 characters) | The name of the user completing the account activity |
| <logentry> | String  (300 characters) | The account activity to be logged |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>logAccountActivity</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<logentry>ip address added</logentry>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - logoutUser

Use the logoutUser command to terminate the current session for the specified user. My Sage Pay will automatically time-out user accounts after 15 minutes of inactivity, but this command can be used to force a logout if necessary.

 The user name provided must have administrative privileges to run this command.

See also: createUser, deleteUser, setUserRights, lockUser, unlockUser, setUserPassword

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  logoutUser | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user account to log out. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML input:**

<vspaccess>

<command>logoutUser</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Response XML Output:**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - unlockUser

Use this command to unlock user accounts that have been locked out either by three incorrect login attempts, or by the account being specifically locked by an administrator.

 The user name provided must have administrative privileges to run this command.

See also: createUser, deleteUser, setUserRights, lockUser, logoutUser, setUserPassword

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  unlockUser | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user account to unlock. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>unlockUser</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setUserPassword

This command can be used to overwrite the existing user’s password with the new password specified. This is useful if the user has forgotten their password.

 The user name provided must have administrative privileges to run this command.

See also: createUser, deleteUser, setUserRights, lockUser, unlockUser, logoutUser

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setUserPassword | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user account requiring the new password. The user name cannot be the same as the vendor (i.e. it cannot be the admin account) |
| <password> | String  (32 characters) | The new user’s password. CASE SENSITIVE and can contain ASCII characters |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setUserPassword</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<password>password123</password>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setUserRights

The setUserRights command allows you to overwrite the privileges assigned to the given My Sage Pay and Reporting & Admin API user. This is equivalent to editing the user in My Sage Pay. NB: This command will REMOVE ALL EXISTING access rights and replace them with those specified. It does NOT add to the existing rights.

 The user name provided must have administrative privileges to run this command.

See also: createUser, deleteUser, lockUser, unlockUser, logoutUser, setUserPassword

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setUserRights | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user account requiring the access right changes. The user name cannot be the same as the vendor (i.e. it cannot be the admin account) |
| <accessrights> | Node | Container for the access rights |
| └<access> | String  (specific values) | Access values can be:  **Transactions** – Access to the transaction area  **Reports** – Access to the reporting area  **Terminal** – Can process Terminal transactions  **Admin** – USER HAS ADMIN RIGHTS!  **CanViewAll** – Can view other user’s transactions as well as their own.  **CanRefund**, **CanRelease**, **CanAbort**, **CanRepeat**, **CanVoid** and **CanManual** – user can perform these transaction types if passed |
| <homepage> | String  (20 characters) | The initial page the user is presented with when they log in. The following values can be used:  **transactions.msp** – the report generation page  **settings.msp** – the account administration page  **terminal.msp** – the virtual terminal page  **search.msp** – search transactions |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setUserRights</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<accessrights>

<access>Terminal</access>

</accessrights>

<homepage>terminal.msp</homepage>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - set3DSecureStatus

This command allows you to switch 3D Secure checks on or off for the chosen vendor. You cannot remove or add rules with this command, so switching on 3D Secure will also turn on existing 3D Secure rules (if they exist on this account)

 The user name provided must have administrative privileges to run this command.

See also: getAVSCV2Status, setAVSCV2Status, get3DSecureStatus

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  set3DSecureStatus | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <status> | String  (3 characters) | Should contain either **ON** or **OFF** to indicate the preferred status of the 3D Secure checks. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Output**

<vspaccess>

<command>set3DSecureStatus</command>

<vendor>vendorname</vendor>

<user>username</user>

<status>ON</status>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setAVSCV2Status

This command allows you to switch AVSCV2 checks on or off for the chosen vendor. You cannot remove or add rules with this command, so switching on AVSCV2 will also turn on existing AVSCV2 rules (if they exist on this account).

 The user name provided must have administrative privileges to run this command.

See also: get3DSecureStatus, getAVSCV2Status, set3DSecureStatus

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setAVSCV2Status | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <status> | String  (3 characters) | Should contain either **ON** or **OFF** to indicate the preferred status of the AVSCV2 checks. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setAVSCV2Status</command>

<vendor>vendorname</vendor>

<user>username</user>

<status>ON</status>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - set3DSecureRules

This command sets up a rule set for the 3D Secure authentication system.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  set3DSecureRules | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <startvalue> | Numeric  (monetary value) | The starting value you wish the particular rule to apply from (must not overlap existing rules) |
| <endvalue> | Numeric  (monetary value) | The finishing value you wish the particular rule to apply to (must not overlap existing rules) |
| <perform3dauth> | String  (3 characters) | **YES** or **NO** |
| <allownon3dcards> | String  (3 characters) | **YES** or **NO** |
| <allownon3dissuers> | String  (3 characters) | **YES** or **NO** |
| <allow3dfailures> | String  (3 characters) | **YES** or **NO** |
| <allowmpifailures> | String  (3 characters) | **YES** or **NO** |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>set3DSecureRules</command>

<vendor>vendorname</vendor>

<user>username</user>

<startvalue>0.00</startvalue>

<endvalue>1000.00</endvalue>

<perform3dauth>YES</perform3dauth>

<allownon3dcards>YES</allownon3dcards>

<allownon3dissuers>YES</allownon3dissuers>

<allow3dfailures>YES</allow3dfailures>

<allowmpifailures>YES</allowmpifailures>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setAVSCV2Rules

This command sets up a rule set for the AVSCV2 authentication system.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setAVSCV2Rules | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <startvalue> | Numeric  (monetary value) | The starting value you wish the particular rule to apply from (must not overlap existing rules) |
| <endvalue> | Numeric  (monetary value) | The finishing value you wish the particular rule to apply to (must not overlap existing rules) |
| <allownodatamatches> | String  (3 characters) | **YES** or **NO** |
| <allowaddressmatchonly> | String  (3 characters) | **YES** or **NO** |
| <allowsecuritycodematchonly> | String  (3 characters) | **YES** or **NO** |
| <allowdatanotchecked> | String  (3 characters) | **YES** or **NO** |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setAVSCV2Rules</command>

<vendor>vendorname</vendor>

<user>username</user>

<startvalue>0.00</startvalue>

<endvalue>1000.00</endvalue>

<allownodatamatches>YES</allownodatamatches>

<allowaddressmatchonly>YES</allowaddressmatchonly>

<allowsecuritycodematchonly>YES</allowsecuritycodematchonly>

<allowdatanotchecked>YES</allowdatanotchecked>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setExpiryDate

This command allows the expiry date of the card used in the specified transaction, to be changed. This is useful for continuous authority payments where cards have been re-issued but the card number has not been changed. This modified transaction can then be REPEATed against without authorisation failures.

The transactions can be specified by either VendorTxCode or VPSTxID. The new ExpiryDate must be in MMYY (two digit month and two digit year) format.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setExpiryDate | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <vendortxcode>  (optional) **OR** | String  (40 characters) | The VendorTxCode of the transaction |
| <vpstxid>  (optional) | String  (40 characters) | The VPSTxID (Transaction ID) of the transaction |
| <startdate>  (optional) | String  (4 characters) | MMYY format e.g. 0307 for March 2007 |
| <expirydate> | String  (4 characters) | MMYY format e.g. 0309 for March 2009 |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input 1**

<vspaccess>

<command>setExpiryDate</command>

<vendor>vendorname</vendor>

<user>username</user>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<expirydate>0114</expirydate>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Input 2**

<vspaccess>

<command>setExpiryDate</command>

<vendor>vendorname</vendor>

<user>username</user>

<vpstxid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vpstxid>

<expirydate>0114</expirydate>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Input 3**

<vspaccess>

<command>setExpiryDate</command>

<vendor>vendorname</vendor>

<user>username</user>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<startdate>0108</ startdate >

<expirydate>0114</expirydate>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setVendorDetails

Using this command will set the vendor’s account details.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setVendorDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <vendordetails> | Node | A container for the vendor details elements. |
| └<vendorprovidedname>  (optional) | String  (50 characters) | The actual name of the vendor. |
| └<homepageurl>  (optional) | String  (255 characters) | The website address for the vendor account.The protocol must be included for the URL to be valid |
| └<supportemailaddress>  (optional) | String  (100 characters) | The support email address for the vendor account. |
| └<sendhtmlemails>  (optional) | String  (3 characters) | Either **YES** if emails will be sent with HTML formatting, **NO** if not. |
| └<defaultterminalcurrency>  (optional) | String  (3 characters) | The default currency for terminal transactions. Must conform to the ISO 4217 3-letter currency code of a supported currency. If it doesn’t conform, then the currency field is not updated |
| └<defaultlanguage>  (optional) | String  (2 characters) | The default language for the payment pages. Must conform to the ISO 639-2 2-letter language code.  If the language is not supported then the language supported in the templates will be used  Currently supported languages in the Default templates are : French, German, Spanish, Portuguese, Dutch and English |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setVendorDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<vendordetails>

<vendorprovidedname>My Company Name</vendorprovidedname>

<homepageurl>http://www.yourbusinesshomepage.com</homepageurl>

<supportemailaddress>name@email.com</supportemailaddress>

<sendhtmlemails>YES</sendhtmlemails>

<defaultterminalcurrency>GBP</defaultterminalcurrency>

<defaultlanguage>DE</defaultlanguage>

</vendordetails>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command – setUserDetails

Using this command will set the vendor’s account details.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setUserDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | String  (20 characters) | Username that will be updated |
| <firstname>  (optional) | String  (70 characters) | The first name of the user |
| <lastname>  (optional) | String  (70 characters) | The last name of the user |
| <emailaddress>  (optional) | String  (255 characters) | Email address to which all notifications will be sent |
| <commsoptin>  (optional) | String  (3 characters) | Either **YES** to opt into to emails from the Sage Pay team, **NO** if not. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setUserDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>myuser</username>

<emailaddress>myuser@mycompany.com</emailaddress>

<firstname>myfirstname</firstname>

<lastname>mylastname</lastname>

<commsoptin>YES</commsoptin>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - addBlockedCardRanges

This command adds card BIN ranges to the blocked card range list on the vendor account. By adding these BIN ranges, transactions will then be blocked from customers with cards within these ranges.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  addBlockedCardRanges | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedcardranges> | node | Node containing country code elements |
| └<blockedcardrange> | Numeric  (9 characters) | The card BIN range to be blocked |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addBlockedCardRanges</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedcardranges>

<blockedcardrange>675967511</blockedcardrange>

</blockedcardranges>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - addBlockedCountries

This command adds country codes to the blocked countries list on the vendor account. By adding these countries, transactions will then be blocked from customers within these countries.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  addBlockedCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedcountries> | node | Node containing country code elements |
| └<countrycode> | String  (2 characters) | ISO3166 2 character country code |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addBlockedCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedcountries>

<countrycode>AF</countrycode>

<countrycode>AL</countrycode>

</blockedcountries>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - addBlockedIPs

This command adds selected blocked IP addresses and subnet masks to the blocked IP addresses list on the vendor account. By adding these IP addresses from the list, transactions will no longer be allowed from customers with these IP addresses.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  addBlockedIPs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedips> | node | Node containing the IP address nodes |
| └<ipaddress> | Node | Node containing the IP address and subnet mask values |
| └<address> | String  (15 characters) | The IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<mask> | String  (15 characters) | The subnet mask for the IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addBlockedIPs</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedips>

<ipaddress>

<address>123.123.123.123</address>

<mask>255.255.255.255</mask>

</ipaddress>

</blockedips>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - addBlockedIssuingCountries

This command adds a country to the list of blocked card issuing countries. Customers with cards issued from these countries will be prevented from processing transactions against the provided vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  addBlocked  IssuingCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor account against which this command will be executed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedissuingcountries> | Node | Node containing country code elements |
| L<countrycode> | String  (2 characters) | Must be an ISO 2 character country code. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addBlockedIssuingCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedissuingcountries>

<countrycode>AF</countrycode>

<countrycode>AL</countrycode>

</blockedissuingcountries>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - addValidIPs

This command will add requested valid IP addresses and subnet masks to the valid IP addresses list on the vendor account. This list of valid IP addresses is used to validate where the transaction is being created from for Server and Direct transactions.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  addValidIPs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <validips> | node | Node containing the IP address nodes |
| └<ipaddress> | Node | Node containing the IP address and subnet mask values |
| └<address> | String  (15 characters) | The IP address to be allowed in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<mask> | String  (15 characters) | The subnet mask for the IP address to be allowed in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<note>  (optional) | String  (255 characters) | A description of the IP address |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addValidIPs</command>

<vendor>vendorname</vendor>

<user>username</user>

<validips>

<ipaddress>

<address>123.123.123.123</address>

<mask>255.255.255.255</mask>

<note>my ip address</note>

</ipaddress>

</validips>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - addRestrictedDomains

This command will add a restricted domain to the vendor account. When restricted domain(s) are added to the account then User’s may only have email addresses from this domain(s).

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  addRestrictedDomains | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <restricteddomains> |  | Node to hold the restricted domain |
| └<restricteddomain> | String(70) | The restricted domain. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addRestrictedDomain</command>

<vendor>vendorname</vendor>

<user>username</user>

<restricteddomains>

<restricteddomain>restricted.co.uk</restricteddomain>

</restricteddomains>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setVendorSurcharges

This command allows a Vendor to add and amend multiple default surcharges to their account

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setVendorSurcharges | The service being requested. |
| <vendor> | String  (15 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <vendorsurcharges> | node | Node containing the Surcharges |
| └<vendorsurcharge> | Node | Node containing the surcharge details to be added |
| └<paymentsystem> | String  (12 characters) | The payment methoe used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MCDEBIT** (MasterCard Debit),  **PAYPAL, SWITCH** (UK Maestro)**, UKE** (Visa Electron)or **VISA** (Visa Credit)  For European Payments **EPS**, **GIROPAY**,**IDEAL**, **SOFORT**, **ELV** |
| └<currency> | String  (3 characters) | Must conform to the ISO 4217 3-letter currency code of a supported currency. |
| └<surchargetype> | String  (10 characters) | The type of surcharge being applied.  Contains either  **FIXED** or **PERCENTAGE** |
| └<surcharge > | Numeric  (decimal) | The percentage or fixed amount to be charged as a default |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>addVendorSurcharges</command>

<vendor>vendorname</vendor>

<user>username</user>

< vendorsurcharges>  
< vendorsurcharge>  
< paymentsystem>MC</paymentsystem>  
< currency>GBP</currency>  
< surcharge>2.1</surcharge>  
< surchargetype>PERCENTAGE</surchargetype>  
< /vendorsurcharge>  
< /vendorsurcharges>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteAVSCV2Rules

This command removes all existing AVSCV2 rules on the vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteAVSCV2Rules | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteAVSCV2Rules</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - delete3DSecureRules

This command removes all existing 3D Secure rules on the vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  delete3DSecureRules | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>delete3DSecureRules</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteBlockedCardRanges

This command deletes card BIN ranges from the blocked card range list on the vendor account. By removing these BIN ranges, transactions will be allowed from customers with cards within these ranges.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteBlockedCardRanges | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedcardranges> | node | Node containing country code elements |
| └<blockedcardrange> | Numeric  (9 characters) | The card BIN range to be blocked |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteBlockedCardRanges</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedcardranges>

<blockedcardrange>675967511</blockedcardrange>

<blockedcardrange>688968686</blockedcardrange>

</blockedcardranges>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteBlockedCountries

This command removes country codes which are currently in the blocked countries list on the vendor account. By deleting these countries, transactions will then be allowed from customers within these countries.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteBlockedCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedcountries> | node | Node containing country code elements |
| └<countrycode> | String  (2 characters) | ISO3166 2 character country code |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteBlockedCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedcountries>

<countrycode>UK</countrycode>

</blockedcountries>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteBlockedIPs

This command deletes selected IP addresses and subnet masks from the blocked IP addresses list on the vendor account. By deleting these IP addresses from the list, transactions will be allowed from customers with these IP addresses.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteBlockedIPs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedips> | node | Node containing the IP address nodes |
| └<ipaddress> | Node | Node containing the IP address and subnet mask values |
| └<address> | String  (15 characters) | The IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<mask> | String  (15 characters) | The subnet mask for the IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteBlockedIPs</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedips>

<ipaddress>

<address>123.123.123.123</address>

<mask>255.255.255.255</mask>

</ipaddress>

</blockedips>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteBlockedIssuingCountries

This command deletes selected countries from the blocked issuing countries list on the vendor account. By deleting these countries from the list, transactions will be allowed from customers with cards issued from these countries.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteBlockedIssuingCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor account against which this command will be executed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <blockedissuingcountries> | Node | Node containing country code elements |
| L<countrycode> | String  (2 characters) | Must be an ISO 2 character country code. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteBlockedIssuingCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<blockedissuingcountries>

<countrycode>AZ</countrycode>

</blockedissuingcountries>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteValidIPs

This command removes selected valid IP addresses and subnet masks which are currently in the valid IP addresses list on the vendor account. This list of valid IP addresses is used to validate where the transaction is being created from for Server and Direct transactions.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteValidIPs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <validips> | node | Node containing the IP address nodes |
| └<ipaddress> | Node | Node containing the IP address and subnet mask values |
| └<address> | String  (15 characters) | The IP address to be allowed in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<mask> | String  (15 characters) | The subnet mask for the IP address to be allowed in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteValidIPs</command>

<vendor>vendorname</vendor>

<user>username</user>

<validips>

<ipaddress>

<address>123.123.123.123</address>

<mask>255.255.255.255</mask>

</ipaddress>

</validips>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setPaymentPageTemplates

This command selects the type of payment page you would like to display to your customer. This is only available to merchants processing through Form or Server.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setPaymentPageTemplates | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <templatetype> | String  (17 characters) | The type of payment page template you would like displaying to your customers:  **Default**: The standard Sage Pay templates with editable address field  **DefaultAddrRDOnly**:The standard Sage Pay templates with read-only address fields  **DefaultNoAddr**: The standard Sage Pay templates with no visible address fields  **Custom**: Your own custom template design (previously submitted to Sage Pay by you) |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setPaymentPageTemplates</command>

<vendor>vendorname</vendor>

<user>username</user>

<templatetype>default</templatetype>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - setPaypalAccount

The setPaypalAccount command sets the requested PayPal account. It is a two step process involving contacting PayPal to verify the email address exists.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  setPaypalAccount | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <paypaluserid> | String  (100 characters) | The PayPal account to be set |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>setPaypalAccount</command>

<vendor>vendorname</vendor>

<user>username</user>

<paypaluserid>name@email.com</paypaluserid>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deletePaypalAccount

The deletePaypalAccount command deletes the requested PayPal account from the vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deletePaypalAccount | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deletePaypalAccount</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteRestrictedDomain

This command removes a restricted domain from a vendor account

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteRestrictedDomain | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <restricteddomains> | Node | Node to hold domains |
| <restricteddomain> | String  (70 characters) | The domain to be deleted. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteRestrictedDomain</command>

<vendor>vendorname</vendor>

<user>username</user>

<restricteddomains>

<restricteddomain>restricted.co.uk</restricteddomain>

</restricteddomains>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Admin Command - deleteVendorSurcharges

This command allows a Vendor to Delete multiple default surcharges from their account

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  deleteVendorSurcharges | The service being requested. |
| <vendor> | String  (15 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <vendorsurcharges> | node | Node containing the Surcharges |
| └<vendorsurcharge> | Node | Node containing the surcharge details to be deleted |
| <paymentsystem> | String  (12 characters) | The payment method used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MCDEBIT** (MasterCard Debit),  **PAYPAL, SWITCH** (UK Maestro)**, UKE** (Visa Electron)or **VISA** (Visa Credit)  For European Payments **EPS**, **GIROPAY**,**IDEAL**, **SOFORT**, **ELV** |
| <currency> | String  (3 characters) |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |

**Sample XML Input**

<vspaccess>

<command>deleteVendorSurcharges</command>

<vendor>vendorname</vendor>

<user>username</user>

< vendorsurcharges>  
< vendorsurcharge>  
< paymentsystem>DC</paymentsystem>  
< currency>GBP</currency>   
< /vendorsurcharge>  
< vendorsurcharge>  
< paymentsystem>IDEAL</paymentsystem>  
< currency>HKD</currency>  
< /vendorsurcharge>  
< /vendorsurcharges>  
<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - doesUserExist

The doesUserExist command returns whether the user exists on the account of the provided vendor name.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  doesUserExist | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | Alphanumeric  (20 characters) | The user name. Can only contain letters and numbers. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <vendorid> | String  (40 characters) | The VendorId of the user (same as the vendor account) |
| <username> | String  (20 characters) | The name of the user. |
| <userexists> | String  (3 characters) | Whether the user exists - either **YES** or **NO** |

**Sample XML Input**

<vspaccess>

<command>doesUserExist</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<vendorid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vendorid>

<username>username2</username>

<userexists>YES</userexists>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getUserRights

This command returns a list of access privileges set on the provided user name detailing their access to the vendor’s account and transaction information.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getUserRights | The service being requested. |
| <vendor> | String  (15 characters) | The vendor account on which the user is present. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <username> | String  (20 characters) | The user who’s access rights will be returned. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <vendorid> | String  (40 characters) | The vendorID of the vendor account the user is associated with. |
| <username> | String  (20 characters) | The user who’s access rights will be returned. |
| <admin> | String  (3 characters) | **YES** or **NO** |
| <islocked> | String  (3 characters) | **YES** or **NO** |
| <reports> | String  (3 characters) | **YES** or **NO** |
| <terminal> | String  (3 characters) | **YES** or **NO** |
| <transactions> | String  (3 characters) | **YES** or **NO** |
| <updates> | String  (3 characters) | **YES** or **NO** |
| <canabort> | String  (3 characters) | **YES** or **NO** |
| <canmanual> | String  (3 characters) | **YES** or **NO** |
| <canrefund> | String  (3 characters) | **YES** or **NO** |
| <canrelease> | String  (3 characters) | **YES** or **NO** |
| <canrepeat> | String  (3 characters) | **YES** or **NO** |
| <canviewall> | String  (3 characters) | **YES** or **NO** |
| <canvoid> | String  (3 characters) | **YES** or **NO** |
| <loggedin> | String  (3 characters) | **YES** or **NO** |
| <homepage> | String  (20 characters) | The page which the user will initially be logged into |

**Sample XML Input**

<vspaccess>

<command>getUserRights</command>

<vendor>vendorname</vendor>

<user>username</user>

<username>username2</username>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<vendorid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vendorid>

<username>username2</username>

<admin>YES</admin>

<islocked>NO</islocked>

<reports>YES</reports>

<terminal>YES</terminal>

<transactions>YES</transactions>

<updates>YES</updates>

<canabort>YES</canabort>

<canmanual>YES</canmanual>

<canrefund>YES</canrefund>

<canrelease>YES</canrelease>

<canrepeat>YES</canrepeat>

<canviewall>YES</canviewall>

<canvoid>YES</canvoid>

<loggedin>YES</loggedin>

<homepage>search.msp</homepage>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getAccountLogs

This command retrieves the account log detailing all the changes to the Reporting database.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getAccountLogs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <logentries> | Node | Node containing all log entries |
| └<logrow> | Node | Node containing individual log entry |
| └<logdate> | Datetime  (DD/MM/YYYY HH:MM:SS:sss) | The date and time of the log entry |
| └<username> | String  (20 characters) | The name of the user who completed the activity |
| └<logentry> | String  (300 characters) | The log entry details |

**Sample XML Input**

<vspaccess>

<command>getAccountLogs</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<logentries>

<logrow>

<logdate>08/07/2010 15:42:22.207</logdate>

<logentry>IP added by username2</logentry>

</logrow>

</logentries>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getVendorUserList

This command will obtain a list of all users on the vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getVendorUserList | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <users> | node | The node containing the returned users. |
| L <user> | node | The node representing the user. |
| L<username> | String  (20 characters) | The name of the user. |
| L<islocked> | String  (Boolean) | Will contain either **true** or **false** to indicate whether the user account is locked from failed login attempts |
| L<lastactivity>  (optional) | Datetime  (dd/mm/yyyy hh:mm:ss) | A Timestamp indicating the last action of the user within the Sage Pay systems if they have an open session. |
| L<admin> | String  (Boolean) | Will contain either **true** or **false** to indicate whether the user account is given Admin privileges in MSP. |
| L<transactions> | String  (Boolean) | **True** or **false** dependant on the user’s access right for the transaction section. |
| L<reports> | String  (Boolean) | **True** or **false** dependant on the user’s access right for the reporting section. |
| L<terminal> | String  (Boolean) | **True** or **false** dependant on the user’s access right for the terminal section. |
| L<updates> | String  (Boolean) | **True** or **false** dependant on the user’s access right for the updates section. |
| L<canviewall> | String  (Boolean) | **True** or **false** dependant on the user’s access right to view all types of transaction in My Sage Pay |
| L<canrefund> | String  (Boolean) | **True** or **false** dependant on the user’s access right to process refunds of transactions in My Sage Pay |
| L<canrelease> | String  (Boolean) | **True** or **false** dependant on the user’s access right to process releases of transactions in My Sage Pay |
| L<canrepeat> | String  (Boolean) | **True** or **false** dependant on the user’s access right to process repeats of transactions in My Sage Pay |
| L<canabort> | String  (Boolean) | **True** or **false** dependant on the user’s access right to process aborts of transactions in My Sage Pay |
| L<canvoid> | String  (Boolean) | **True** or **false** dependant on the user’s access right to process voiding of transactions in My Sage Pay |
| L<canmanual> | String  (Boolean) | **True** or **False** dependant on the user’s access right to process manual transactions in My Sage Pay |
| L<vendorid> | String  (40 characters) | The VendorID of the user (same as the vendor account) |
| L<homepage> | String (30 characters) | The landing page which the user will be presented with when logging into My Sage Pay. |

**Sample XML Input**

<vspaccess>

<command>getVendorUserList</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<users>

<user>

<username>username2</username>

<islocked>true</islocked>

<lastactivity>20/10/2010 15:51:12.927</lastactivity>

<admin>true</admin>

<transactions>true</transactions>

<reports>false</reports>

<terminal>true</terminal>

<updates>false</updates>

<canviewall>false</canviewall>

<canrefund>false</canrefund>

<canrelease>false</canrelease>

<canrepeat>false</canrepeat>

<canabort>false</canabort>

<canvoid>false</canvoid>

<canmanual>false</canmanual>

<vendorid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vendorid>

<homepage>settings.msp</homepage>

</user>

</users>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getVendorDetails

This command will obtain the details of the provided vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <vendorprovidedname> | String  (50 characters) | The vendors provided name. |
| <homepageurl> | String  (255 characters) | The vendors website URL. |
| <fromemail> | String  (100 characters) | The vendors registered email address for contact by customers and Sage Pay support. |
| <sendhtmlemails> | String  (3 characters) | Will contain either **YES** or **NO** to indicate whether HTML emails are being sent. |
| <defaultterminalcurrency>  (optional) | String  (3 characters) | The default transaction currency for the provided vendor as an ISO 4217 currency code. |
| <logo> | String  (255Characters) | The filename name of the current logo on the account |
| <isgiftaid> | String  (3 characters) | Will contain either **YES** or **NO** to indicate the vendor account is set up for Gift Aid |
| <isavscv2> | String  (3 characters) | Will contain either **YES** or **NO** to indicate the vendor account is set up for AVSCV2 |
| <isthreed> | String  (3 characters) | Will contain either **YES** or **NO** to indicate the vendor account is set up for 3D Secure |
| <terminalonly> | String  (3 characters) | Will contain either **YES** or **NO** to indicate the vendor account is terminal only |
| <directrefund> | String  (3 characters) | Will contain either **YES** or **NO** to indicate the vendor account has direct refund |
| <paypaluserid> | String  (100 characters) | The PayPal user id. |
| <templatetype> | String  (9 characters) | The type of payment page template currently displayed to your customers:  **Default**: The standard Sage Pay templates with editable address field  **DefaultAddrRDOnly**:The standard Sage Pay templates with read-only address fields  **DefaultNoAddr**: The standard Sage Pay templates with no visible address fields  **Custom**: Your own custom template design (previously submitted to Sage Pay by you) |
| <pproenabled> | String  (3 characters) | Will contain either **YES** or **NO** to indicate the vendor account is PPro enabled. |
| <defaultlanguage> | String  (2 characters) | The language code for the vendor to be passed per transaction so the payment pages are displayed in the preferred language. |

**Sample XML Input**

<vspaccess>

<command>getVendorDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<vendorprovidedname>my company name</vendorprovidedname>

<homepageurl>http://www.webaddress.com</homepageurl>

<fromemail>name@email.com</fromemail>

<sendhtmlemails>YES</sendhtmlemails>

<defaultterminalcurrency>GBP</defaultterminalcurrency>

<logo>mylogo.gif</logo>

<isclosed></isclosed>

<isgiftaid>NO</isgiftaid>

<isavscv2>NO</isavscv2>

<isthreed>NO</isthreed>

<terminalonly>NO</terminalonly>

<directrefund>NO</directrefund>

<paypaluserid>name@email.co.uk</paypaluserid>

<templatetype>noaddress</templatetype>

<pproenabled>YES<pproenabled>

<defaultlanguage>DE<defaultlanguage>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getVendorPProDetail

Using this command will obtain the details of the PPro account

 This command doesn’t require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getVendorPProDetail | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <pproaccount> | Node | Container element for other details. |
| L<contractId> | String | Contract id for the ppro account |
| L<channelId> | String | Channel id for the ppro account |
| L< active> | true/false | True if active else false. |
| L<pproTags> | Node | Container for tag elements |
| L<pproTag> | Node | Container for tag details |
| L< tag> | String | Tag value provided. |
| L<paymentsystem> | String | PPro Payment system name. **EPS, GIROPAY, IDEAL, ELS,SOFORT** |
| L< live> |  | **True** or **False** |

**Sample XML Input**

<vspaccess>

<command>getVendorPProDetail</command>

<vendor> mycompanyname </vendor>

<user> myusername </user>

<signature>F7414824C864FE359249DD5C911E64B2</signature>

</vspaccess>

**Sample XML Output**

<?xml version="1.0" encoding="ISO-8859-1" ?>

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>16/05/2012 15:21:41</timestamp>

<pproaccount>

<contractId>contractid</contractId>

<channelId>channel name</channelId>

<active>true</active>

<pproTags>

<pprotag>

<tag>eps</tag>

<paymentsystem>EPS</paymentsystem>

<live>true</live>

</pprotag>

<pprotag>

<tag>giropay</tag>

<paymentsystem>GIROPAY</paymentsystem>

<live>true</live>

</pprotag>

<pprotag>

<tag>ideal</tag>

<paymentsystem>IDEAL</paymentsystem>

<live>true</live>

</pprotag>

</pproTags>

</pproaccount>

</vspaccess>

## Reporting Command - getVendorPaymentSystems

Using this command will obtain the details of the payment systems available to the provided vendor account.

 This command doesn’t require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getVendorPaymentSystems | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <currency>  (optional) | String (3 characters) | ISO-4217 3 character currency code. eg: GBP |
| <accounttype>  (optional) | String  (1 character) | The account type of the payment system. **E**,**C** or **M**. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <paymentsystems> | Node | Node to contain payment systems |
| L<paymentsystem> | String (50 characters) | The name of the payment system (card type) acceptable to this vendor account. |

**Sample XML Input**

<vspaccess>

<command>getVendorPaymentSystems</command>

<vendor>vendorname</vendor>

<user>username</user>

<currency>GBP</currency>

<accounttype>M</accounttype>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<paymentsystems>

<paymentsystem>American Express</paymentsystem>

<paymentsystem>MasterCard</paymentsystem>

<paymentsystem>Maestro</paymentsystem>

<paymentsystem>UK Maestro</paymentsystem>

<paymentsystem>Visa</paymentsystem>

<paymentsystem>Visa Debit / Delta</paymentsystem>

<paymentsystem>Visa Electron</paymentsystem>

</paymentsystems>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getValidIPs

This command gets a list of all valid IPs that are set up on the account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getValidIPs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <ipaddresses> | Node | Node containing the IP address nodes |
| └<ipaddress> | Node | Node containing the IP address and subnet mask values |
| └<address> | String  (15 characters) | The IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<mask> | String  (15 characters) | The subnet mask for the IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<note> | String  (255 characters) | A description of the IP address |

**Sample XML Input**

<vspaccess>

<command>getValidIPs</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<ipaddresses>

<ipaddress>

<address>123.123.123.123</address>

<mask>255.255.255.255</mask>

<note>Our IP Address</note>

</ipaddress>

</ipaddresses>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getURLs

This command will return the fully qualified service URLs for the transaction types supported by the vendor account. By default, all URLs for all supported transaction types are returned. You can optionally specify the transaction types you wish to receive URLs to, if you wish to filter the returned list.

 This command doesn’t require the user to have administrator rights.

See also: getTransactionTypes

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getURLs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <system> | String  (10 characters) | The system for which the URLs are being requested. Must be either **VSPServer**, **VSPDirect** or **VSPForm** |
| <txtypes>  (optional) | Node | Container for the transaction types being returned |
| └<txtype>  (optional) | String  (20 characters) | The transaction types supported. Contains **PAYMENT**, **REFUND**, **DEFERRED**, **REPEAT**, **REPEATDEFERRED**, **MANUAL**, **AUTHENTICATE**, **TOKEN**, **REMOVETOKEN**, **COMPLETE** |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <urls> | Node | Container for the URLs being returned |
| └<url> | Node | Container for a single URL with the service it matches |
| └<txtype> | String  (20 characters) | The name of the transaction type for which this URL maps |
| └<serviceURL> | String - FQDN  (255 characters) | Fully Qualified Domain Name URL to the service named above. |

**Sample Input XML**

<vspaccess>

<command>getURLs</command>

<vendor>vendorname</vendor>

<user>username</user>

<system>VSPDirect</system>

<txtypes>

<txtype>PAYMENT</txtype>

<txtype>REFUND</txtype>

<txtype>VOID</txtype>

</txtypes>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<urls>

<url>

<txtype>PAYMENT</txtype>

<serviceURL> https://live.sagepay.com/gateway/service/vspdirect-register.vsp</serviceURL>

</url>

<url>

<txtype>REFUND</txtype>

<serviceURL>https://live.sagepay.com/gateway/service/refund.vsp</serviceURL>

</url>

</urls>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - get3DSecureStatus

The get3dSecureStatus command returns information about whether 3D Secure fraud screening is on or off, and whether rules have been applied.

 The user name provided must have administrative privileges to run this command.

See also: getAVSCV2Status, setAVSCV2Status, set3DSecureStatus

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  get3DSecureStatus | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <status> | String  (3 characters) | Contains either **ON** or **OFF** to indicate whether 3D Secure checks are on or off. |
| <rules> | String  (3 characters) | Contains either **ON** or **OFF** to indicate whether a rulebase is applied to the 3D Secure checks. |

**Sample XML Input**

<vspaccess>

<command>get3DSecureStatus</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</ errorcode>

<status>ON</status>

<rules>OFF</rules>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - get3DSecureRules

This command gets a list of the 3D Secure rules on your account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  get3DSecureRules | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <rules> |  | Node containing the rule nodes |
| └<rule> | Node | Node containing the rule values |
| └<startvalue> | Numeric  (monetary value) | The starting value you wish the particular rule to apply from (must not overlap existing rules) |
| └<endvalue> | Numeric  (monetary value) | The finishing value you wish the particular rule to apply to (must not overlap existing rules) |
| └<perform3dauth> | String  (3 characters) | **YES** or **NO** |
| └<allownon3dcards> | String  (3 characters) | **YES** or **NO** |
| └<allownon3dissuers> | String  (3 characters) | **YES** or **NO** |
| └<allow3dfailures> | String  (3 characters) | **YES** or **NO** |
| └<allowmpiauths> | String  (3 characters) | **YES** or **NO** |

**Sample XML Input**

<vspaccess>

<command>get3DSecureRules</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<rules>

<rule>

<startvalue>0.00</startvalue>

<endvalue>1000.00</endvalue>

<perform3dauth>YES</perform3dauth>

<allownon3dcards>YES</allownon3dcards>

<allownon3dissuers>YES</allownon3dissuers>

<allow3dfailures>YES</allow3dfailures>

<allowmpiauths>YES</allowmpiauths>

</rule>

</rules>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getAVSCV2Status

This command returns information about whether AVSCV2 fraud screening is on or off, and whether rules have been applied.

 The user name provided must have administrative privileges to run this command.

See also: get3DSecureStatus, setAVSCV2Status, set3DSecureStatus

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getAVSCV2Status | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <status> | String  (3 characters) | Contains either **ON** or **OFF** to indicate whether AVSCV2 checks are on or off. |
| <rules> | String  (3 characters) | Contains either **ON** or **OFF** to indicate whether a rulebase is applied to the AVSCV2 checks. |

**Sample XML Input**

<vspaccess>

<command>getAVSCV2Status</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<status>ON</status>

<rules>ON</rules>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getAVSCV2Rules

This command gets a list of the AVSCV2 rules on your account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getAVSCV2Rules | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <rules> | node | Node containing rules |
| └<rule> | node | Container for the rule |
| └<startvalue> | Numeric  (monetary value) | The starting value you wish the particular rule to apply from (must not overlap existing rules) |
| └<endvalue> | Numeric  (monetary value) | The finishing value you wish the particular rule to apply to (must not overlap existing rules) |
| └<allownodatamatches> | String  (3 characters) | **YES** or **NO** |
| └<allowaddressmatchonly> | String  (3 characters) | **YES** or **NO** |
| └<allowsecuritycodematchonly> | String  (3 characters) | **YES** or **NO** |
| └<allowdatanotchecked> | String  (3 characters) | **YES** or **NO** |

**Sample XML Input**

<vspaccess>

<command>getAVSCV2Rules</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<rules>

<rule>

<startvalue>1.00</startvalue>

<endvalue>100.00</endvalue>

<allownodatamatches>NO</allownodatamatches>

<allowaddressmatchonly>YES</allowaddressmatchonly>

<allowsecuritycodematchonly>NO</allowsecuritycodematchonly>

<allowdatanotchecked>NO</allowdatanotchecked>

</rule>

</rules>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getRestrictedDomains

This command gets a list of the restricted domains associated with an account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getRestrictedDomains | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <restricteddomains> | node | Node containing restricted domains. |
| └< restricteddomain > | String  (70 characters) | A restricted domain |

**Sample XML Input**

<vspaccess>

<command>getRestrictedDomains</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<restricteddomains>

<restricteddomain>resrtricted.co.uk</restricteddomain>

<restricteddomain>resrtricted.com</restricteddomain>

<restricteddomain>resrtricted.org</restricteddomain>

</restricteddomains>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getBatchDetail

This command returns a list of all transactions sent to the bank in the specified settlement file (batch). This command is equivalent to the mySagePay Batch Detail Report (but unlike the report, transactions are not split into payments and refunds.)

 This command doesn’t require the user to have administrator rights.

See also: getTransactionSummary, getTransactionDetail, getBatchList

NOTE: This command returns a maximum of 50 rows of information. The <startrow> and <endrow> fields show which rows have been returned, with <totalrows> showing the total amount of information available.

You can make a request to return specific subsets of the information available by setting the <startrow> and <endrow> values in your request. So for example, if there are 140 rows of information in the batch, your first request will return rows 1 to 50. You can make two subsequent requests to retrieve rows 51 to 100, and 101 to 140 by setting the start and end rows accordingly.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getBatchDetail | The service being requested. |
| <vendor> | String  (16 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <batchid> | Numeric  (Integer) | The unique ID number of this batch for this acquirer |
| <authprocessor> | String  (50 characters) | The name of the acquirer to whom this batch was sent |
| <startrow> (optional) | Integer | The first row of information you require. If left blank this defaults to the first row. |
| <endrow> (optional – must be less than 50 if the startrow is not provided) | Integer | The last row of information you require. By default this is set to 50. A smaller window can be specified. |

Response overleaf

**Response** (continued overleaf)

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <transactions> | Node | A container for the list of transactions returned |
| └<totalrows> | Integer | The total number of rows present. |
| └<transaction> | Node | A container for the individual transaction |
| └<rownumber> | Integer | The current row of information returned. |
| └<startrow> | Integer | The first row of information returned. |
| └<endrow> | Integer | The last row of information returned. |
| └<vpstxid> | String  (40 characters) | The unique VPSTxId (transaction id) of this transaction |
| └<vendortxcode> | String  (40 characters) | The unique VendorTxCode of this transaction |
| └<transactiontype> | String  (15 characters) | The type of transaction. Values are: Authenticate, Authorise, Deferred, Manual, Payment, PreAuth, Refund, Repeat and RepeatDeferred |
| └<systemused> | String  (1 character) | Contains **F** for Form, **S** for Server, **D** for Direct or **T** for Terminal. |
| └<amount> | Numeric  (monetary value) | The value of the transaction ( including any surcharge levied) |
| └<currency> | String  (3 characters) | The ISO 4217 currency code of this transaction |
| └<started> | Datetime  (dd/mm/yyyy hh:mm:ss.SSS) | The date/time that the transaction was registered with Sage Pay |
| └<vpsauthcode>  (optional) | Numeric  (big integer value) | The unique Sage Pay auth code for a successful transaction only |
| └<bankauthcode>  (optional) | String  (20 characters) | The authorisation code from the bank. For PayPal transactions this is the PayPal transaction ID |
| └<cv2result>  (optional) | String  (12 characters) | The results of the CV2 fraud checks.  Either MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED or PARTIAL |
| └<addressresult>  (optional) | String  (12 characters) | The results of the AVS address fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| └<postcoderesult>  (optional) | String  (12 characters) | The results of the AVS postcode fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| └<threedresult>  (optional) | String  (12 characters) | The results of the 3D-Secure fraud checks.  Either OK, NOTCHECKED, ATTEMPTONLY, NOTAVAILABLE, NOTAUTHED, INCOMPLETE or ERROR |
| └<cardholder>  (optional) | String  (100 characters) | If available, the name on the credit/debit card. |
| └<released>  (optional) | Datetime | The date/time this transaction was released, if applicable |
| └<refunded>  (optional) | String  (8 characters) | Either **YES** if fully refunded, **NO** if not refunded or **PARTIAL** if partly refunded |
| └<paymentsystem>  (optional) | String  (20 characters) | The payment system used for the transaction |
| └<repeated>  (optional) | String  (3 characters) | Either **YES** if at least one REPEAT has occurred against this transaction, **NO** if none have |
| └<username>  (optional) | String  (20 characters) | If available, the name of the user account in MySagePay who processed this transaction. |
| └<t3mscore>  (optional) | Numeric | A value in the range -100 to +100 as a fraud score from the third man, when available. |
| └<t3maction>  (optional) | String  (10 characters) | Either **OK, HOLD**, **REJECT** or **NORESULT** as a recommended course of action based on the fraud score. |
| └<t3mid>  (optional) | String  (20 characters) | The ID of this transaction in the 3rd Man systems. |
| └<fraudscreenrecommendation>  (optional) | String (10 characters) | Fraud screening recommendation for the transaction.  Contains either **ACCEPT**, **DENY, CHALLENGE** or **NOTCHECKED** |
| └<fraudcode>  (optional) | String (10 characters) | The fraud provider’s response code. |
| └<fraudcodedetail>  (optional) | String (200 characters) | A human-readable explanation of the fraud code |

**Sample XML Input**

<vspaccess>

<command>getBatchDetail</command>

<vendor>onlinecheese</vendor>

<user>barry</user>

<batchid>123456789</batchid>

<authprocessor>SomeAcquirerXyz</authprocessor>

<startrow>1</startrow>

<endrow>50</endrow>

<signature>799B11DFF4275AEE76531AEC625FADE0</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<transactions>

<transaction>

<rownumber>7</rownumber>

<startrow>1</startrow>

<endrow>7</endrow>

<vpstxid>987A9292-409F-4F23-A4C1-861A78F7C340</vpstxid>

<systemused>T</systemused>

<transactiontype>Payment</transactiontype>

<vendortxcode>T-protx-100524170853-114</vendortxcode>

<started>24/05/2010 17:09:30.333</started>

<amount>23.50</amount>

<currency>GBP</currency>

<vpsauthcode>269956638</vpsauthcode>

<bankauthcode>024576</bankauthcode>

<addressresult>MATCHED</addressresult>

<postcoderesult>MATCHED</postcoderesult>

<threedresult>NOTCHECKED</threedresult>

<cv2result>MATCHED</cv2result>

<cardholder>Greg Cahill V0016487</cardholder>

<refunded>NO</refunded>

<released />

<paymentsystem>Visa Debit / Delta</paymentsystem>

<repeated>NO</repeated>

<username>darren</username>

<t3mscore>-30</t3mscore>

<t3maction>OK</t3maction>

<t3mid>559442606</t3mid>

<fraudscreenrecommendation>DENY</fraudscreenrecommendation>

<fraudcode>0600</fraudcode>

<fraudcodedetail>Card found in stolen card database</fraudcodedetail>

</transaction>

<totalrows>7</totalrows>

</transactions>

<signature>799B11DFF4275AEE76531AEC625FADE0</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<startrow>1</startrow>

<endrow>7</endrow>

<totalrows>7</totalrows>

<transactions>

<transaction>

<rownumber>1</rownumber>

<vpstxid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vpstxid>

<systemused>T</systemused>

<transactiontype>Payment</transactiontype>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<started>24/05/2010 17:09:30.333</started>

<amount>23.50</amount>

<currency>GBP</currency>

<vpsauthcode>999999999</vpsauthcode>

<bankauthcode>123456</bankauthcode>

<addressresult>MATCHED</addressresult>

<postcoderesult>MATCHED</postcoderesult>

<threedresult>NOTCHECKED</threedresult>

<cv2result>MATCHED</cv2result>

<cardholder>CHOlder</cardholder>

<refunded>NO</refunded>

<paymentsystem>Visa Debit / Delta</paymentsystem>

<repeated>NO</repeated>

<released/>

<username>username2</username>

<t3mscore>-30</t3mscore>

<t3maction>OK</t3maction>

<t3mid>123456789</t3mid>

</transaction>

</transactions>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getBatchList

This command returns a list of all bank settlement (batch) files scheduled between the specific dates for the given vendor (equivalent to the Settlement Report in My Sage Pay). Note: The list is limited to 1 month periods to minimise impact on the database.

 This command doesn’t require the user to have administrator rights.

See also: getTransactionList, getTransactionDetail, getBatchDetail

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getBatchList | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <startdate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time from which to begin searching for transactions |
| <enddate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time of the last transaction to be returned. Cannot be more than a month after the start date |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <batches> | Node | Container for the batch information |
| └<batch> | Node | Container for single batch file details |
| └<authprocessor> | String  (50 characters) | The name of the acquirer to whom this batch was sent |
| └<batchid> | Numeric  (Integer) | The unique ID number of this batch for this acquirer |
| └<completed> | Datetime  (dd/mm/yyyy hh:mm:ss.SSS) | The date and time, including milliseconds, the file was submitted to the acquiring bank. |
| └<transactiongroups> | Node | Container for the transaction summary information |
| └<transactiongroup> | Node | Container for the transaction summary information per currency |
| └<currency> | String  (3 characters) | Contains the ISO 4217 3-letter currency code of the batch currency. |
| └<paymentnumber> | Numeric  (Integer) | The number of payment transactions in this batch for this currency |
| └<paymentvalue> | Numeric  (Monetary value) | The total value of the payment transactions in this batch for this currency |
| └<refundnumber> | Numeric  (Integer) | The number of refund transactions in this batch for this currency |
| └<refundvalue> | Numeric  (Monetary value) | The total value of the refund transactions in this batch for this currency |

**Sample XML Input**

<vspaccess>

<command>getBatchList</command>

<vendor>vendorname</vendor>

<user>username</user>

<startdate>01/01/2010 00:00:00</startdate>

<enddate>02/01/2010 00:00:00</enddate>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<batches>

<batch>

<authprocessor>Elavon</authprocessor>

<batchid>1234<batchid>

<completed>01/01/2010 12:00:00</completed>

<transactiongroups>

<transactiongroup>

<currency>GBP</currency>

<paymentnumber>1</paymentnumber>

<paymentvalue>10.00</paymentvalue>

<refundnumber>0</refundnumber>

<refundvalue>0.00</refundvalue>

</transactiongroup>

</transactiongroups>

</batch>

</batches>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getCardType

This command returns the type of card for the first 1-9 digits supplied, allowing you to look up a card type for validation purposes, or correct submission for Direct transactions.

 This command doesn’t require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getCardType | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <cardbegins> | 9-digit numeric | The first 9 digits of the card number must be provided. No spaces or separators must be included. Numbers only |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <paymentsystem> | String  (12 characters) | The card type used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MCDEBIT** (MasterCard Debit)**, SWITCH** (UK Maestro)**, UKE** (Visa Electron)**,** **VISA** (Visa Credit), **LASER** or **UNKNOWN** if card range not recognised |

**Sample XML Input**

<vspaccess>

<command>getCardType</command>

<vendor>vendorname</vendor>

<user>username</user>

<cardbegins>123456789</cardbegins>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>01/03/2012 12:34:56</timestamp>

<paymentsystem>VISA</paymentsystem>

</vspaccess>

## Reporting Command - getCardDetails

This command returns the card details from the first 1-9 digits supplied.

 This command doesn’t require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getCardDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <cardbegins> | 9-digit numeric | The first 9 digits of the card number must be provided . No spaces or separators must be included. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <paymentsystemname> | String  (20 characters) | The payment system (card type) name of the card being used. One of the following:  American Express, Diners Club, JCB, LASER, Maestro, MasterCard, PayPal, UK Maestro, Visa, Visa Debit / Delta, Visa Electron. |
| <shortname> | String  (10 characters) | The card type used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MCDEBIT** (MasterCard Debit)**, SWITCH** (UK Maestro)**, UKE** (Visa Electron)**,** **VISA** (Visa Credit), **LASER** or **UNKNOWN** if card range not recognised |
| <issuedigits> | 1-digit numeric | The number of issue digits for the card |
| <minimum> | 9-digit numeric | The minimum number of the BIN range |
| <maximum> | 9-digit numeric | The maximum number of the BIN range |
| <countrycode> | String  (2 characters) | Contains the ISO 2-letter country code of a country. |
| <issuer> | String  (255 characters) | The card issuer |
| <corporatecard> | String  (3 characters) | Will contain either **YES** or **NO** to indicate whether or not the card is a corporate card |
| <iscredit> | String  (3 characters) | Will contain either **YES** or **NO** to indicate whether or not the card is a credit card |

Example overleaf

**Sample XML Input**

<vspaccess>

<command>getCardDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<cardbegins>540400000</cardbegins>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<paymentsystemname>MasterCard</paymentsystemname>

<shortname>MC</shortname>

<issuedigits>-1</issuedigits>

<minimum>540400000</minimum>

<maximum>540400999</maximum>

<countrycode>RU</countrycode>

<issuer>Joint Stock Commercial Bank “S</issuer>

<corporatecard>NO</corporatecard>

<iscredit>YES</iscredit>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command – getCurrencies

This command returns a list of all currencies supported by the vendor account.

 This command doesn’t require the user to have administrator rights.

See also: getMerchantAccounts

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getCurrencies | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <accounttypes>  (optional) | Node | Container for the account types being requested |
| └<accounttype>  (optional) | String  (1 character) | Provide either **E, C** or **M** to show the account type used in this transaction |
| <paymentsystem>  (optional) | String  (12 characters) | The card type used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MCDEBIT** (MasterCard Debit), **SWITCH** (UK Maestro)**, UKE** (Visa Electron)**,** **VISA** (Visa Credit), **LASER** |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <currencies> | Node | A container for the currency list |
| └<currency> | String  (3 characters) | Contains the ISO 4217 3-letter currency code of a supported currency. |

**Sample XML Input**

<vspaccess>

<command>getCurrencies</command>

<vendor>vendorname</vendor>

<user>username</user>

<accounttypes>

<accounttype>E</accounttype>

</accounttypes>

<paymentsystem>VISA</paymentsystem>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<currencies>

<currency>GBP</currency>

</currencies>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getMerchantAccounts

The getMerchantAccounts command returns a list of the merchant number and bank account details associated with the Sage Pay vendor account. Optionally the request message can be limited to specific currencies and account types, but by default all merchant numbers for each account type (e-commerce, MOTO and continuous authority), for each currency, for each card type will be returned.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getMerchantAccounts | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <currencies>  (optional) | Node | A container for the currency list |
| └<currency>  (optional) | String  (3 characters) | Contains the ISO 4217 3-letter currency code of a supported currency. |
| <accounttypes>  (optional) | Node | A container for account type list |
| └<accounttype>  (optional) | String  (1 character) | Contains either **E**, **C** or **M** to filter the list of returned accounts |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <merchantaccounts> | Node | A container for the merchant number list |
| └<merchantaccount> | Node | Container for this merchant account entry |
| └<merchantnumber> | String  (20 characters) | The merchant number of this account |
| └<authprocessor> | String  (50 characters) | The acquirer who issued the merchant number |
| └<accountnumber> | String  (20 characters) | The bank account associated with this merchant number |
| └<sortcode> | String  (10 characters) | The sort code for the bank account |
| └<accountcurrency> | String  (3 characters) | Contains the ISO 4217 3-letter code of the settlement currency for the bank account |
| └<streamlinenumber>  (optional) | String  (15 characters) | The NatWest Streamline company number for this merchant account (ONLY if acquirer is Streamline) |
| └<currencies>  (optional) | Node | A container for the currency list |
| └<currency>  (optional) | String  (3 characters) | Contains the ISO 4217 3-letter currency code of a supported currency. |
| └<accounttypes>  (optional) | Node | A container for account type list |
| └<accounttype>  (optional) | String  (1 character) | Contains either **E**, **C** or **M** to indicate the accounts types supported on this merchant number. |
| └<paymentsystems>  (optional) | Node | A container for payment system list |
| └<paymentsystem>  (optional) | String  (12 characters) | The card types supported by this account. The card type used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, , MCDEBIT** (MasterCard Debit), **PAYPAL, SWITCH** (UK Maestro)**, UKE** (Visa Electron)**,** **VISA** (Visa Credit), **LASER** |

**Sample XML Input**

<vspaccess>

<command>getMerchantAccounts</command>

<vendor>vendorname</vendor>

<user>username</user>

<currencies>

<currency>GBP</currency>

<currency>EUD</currency>

</currencies>

<accounttypes>

<accounttype>E</accounttype>

<accounttype>M</accounttype>

</accounttypes>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<merchantaccounts>

<merchantaccount>

<merchantnumber>2030405060708090</merchantnumber>

<authprocessor>BARCLAYS MERCHANT SERVICES</authprocessor>

<accountnumber>12312312</accountnumber>

<sortcode>123123</sortcode>

<accountcurrency>GBP</accountcurrency>

<currencies>

<currency>GBP</currency>

<currency>USD</currency>

</currencies>

<accounttypes>

<accounttype>E</accounttype>

<accounttype>M</accounttype>

</accounttypes>

<paymentsystems>

<paymentsystem>AMEX</paymentsystem>

<paymentsystem>MC</paymentsystem>

<paymentsystem>VISA</paymentsystem>

</paymentsystems>

</merchantaccount>

</merchantaccounts>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getTransactionCardDetails

This command returns the last 4 digits of the card number, and the type of card used in the specified successful transaction. Only information about a transaction associated with the vendor can be returned. Transactions can be specified by either VendorTxCode or VPSTxID.

This is a subset of the getTransactionDetails request, to provide often requested functionality.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

See also: getTransactionSummary, getTransactionList, getBatchDetail

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTransactionCardDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <vendortxcode>  (optional) **OR** | String  (40 characters) | The VendorTxCode of the transaction |
| <vpstxid>  (optional) | String  (40 characters) | The VPSTxID (Transaction ID) of the transaction |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <paymentsystem> | String  (12 characters) | The card type used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard),  **MCDEBIT** (MasterCard Debit)**, PAYPAL, SWITCH** (UK Maestro)**, UKE** (Visa Electron)**,** **VISA** (Visa Credit), **LASER** |
| <last4digits> | 4-digit numeric | The last 4 digits of the card number, if available |

**Sample XML Input 1**

<vspaccess>

<command>getTransactionCardDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Input 2**

<vspaccess>

<command>getTransactionCardDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<vpstxid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vpstxid>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<paymentsystem>VISA</paymentsystem>

<last4digits>0000</last4digits>

<timestamp>01/03/2012 12:34:56</timestamp> </vspaccess>

## Reporting Command - getTransactionDetail

This command returns all information held in the database about the specified transaction. Only a transaction associated with the vendor can be returned. Transactions can be specified by either VendorTxCode or VPSTxID.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

See also: getTransactionSummary, getTransactionList, getBatchDetail

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTransactionDetail | The service being requested. |
| <vendor> | String  (16 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <vendortxcode>  (optional) **OR** | String  (40 characters) | The VendorTxCode of the transaction |
| <vpstxid>  (optional) | String  (40 characters) | The VPSTxID (transactionid) of the transaction |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <vpstxid> | String  (40 characters) | The unique VPSTxId (transaction id) of this transaction |
| <vendortxcode> | String  (40 characters) | The unique VendorTxCode of this transaction |
| <transactiontype> | String  (15 characters) | The type of transaction. Values are: Authenticate, Authorise, Deferred, Manual, Payment, PreAuth, Refund, Repeat and RepeatDeferred |
| <txstateid> | Integer | Identifier representing the status of the transaction |
| <status> | String  (255 characters) | A human readable description of the transaction status. |
| <relatedtransactionid>  (optional) | String  (40 characters) | If this transaction is related to another e.g. a REFUND would be related to a parent PAYMENT, this is the vpstxid of that parent |
| <relatedvendortxcode>  (optional) | String  (40 characters) | The VendorTxCode of the related transaction. |
| <description> | String  (200 characters) | The description of good passed at transaction registration |
| <amount> | Numeric  (monetary value) | The value of the transaction (including any surcharge levied) |
| <currency> | String  (3 characters) | The ISO 4217 currency code of this transaction |
| <started> | Datetime | The date/time that the transaction was registered with Sage Pay |
| <completed> | Datetime | The date/time that the transaction was completed |
| <securitykey> | String  (10 characters) | The key used for signing the transaction completion messages |
| <clientip>  (optional) | String  (15 characters) | The IP address of the customer associated with this transaction (where available) |
| <giftaid> | String  (3 characters) | Either **YES** if GiftAid applies to this transaction, or **NO** if it does not |
| <paymentsystem>  (optional) | String  (12 characters) | The card type used in this transaction. Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MCDEBIT** (MasterCard Debit)**, PAYPAL, SWITCH** (UK Maestro)**, UKE** (Visa Electron)or **VISA** (Visa Credit)  For European Payments **EPS**, **GIROPAY**,**IDEAL**, **SOFORT**, **ELV** |
| <paymentsystemdetails>  (optional) | String  (100 characters) | More information about the card used, where available (Issuer, Country etc.) |
| <startdate>  (optional) | String  (4 characters) | The card start date in **MMYY** format, if available |
| <expirydate>  (optional) | String  (4 characters) | The card expiry date in **MMYY** format, if available |
| <last4digits>  (optional) | String  (4 characters) | The last 4 digits of the card number, if available |
| <authprocessor>  (optional) | String  (50 characters) | The name of the acquirer who processed the transaction |
| <merchantnumber>  (optional) | String  (20 characters) | The merchant number through which this transaction passed |
| <accounttype>  (optional) | String  (1 character) | Contains either **E, C** or **M** to show the account type used in this transaction |
| <vpsauthcode>  (optional) | Numeric  (big integer value) | The unique Sage Pay auth code for a successful transaction only |
| <bankauthcode>  (optional) | String  (20 characters) | The authorisation code from the bank. For PayPal transactions this is the PayPal transaction ID |
| <batchid>  (optional) | Numeric  (integer value) | If the transaction has been settled with the bank, this is the ID of that batch |
| <billingfirstnames>  (optional) | String | The customer’s firstnames on the bill. |
| <billingsurname>  (optional) | String | The customer’s surname on the bill. |
| <billingaddress>  (optional) | String  (500 characters) | The customer’s first line of billing address (without the postcode) if available |
| <billingaddress2>  (optional) | String  (500 characters) | The customer’s second line of billing address (without the postcode) if available |
| <billingcity>  (optional) | String | The customer’s billing city if available |
| <billingstate>  (optional) | String | The customer’s billing state if available |
| <billingpostcode>  (optional) | String  (50 characters) | The customer’s billing address post code if available |
| <billingcountry>  (optional) | String | The customer’s billing country |
| <billingphone>  (optional) | String | The customer’s billing phone if possible |
| <deliveryfirstnames>  (optional) | String | The customer’s name for delivery |
| <deliverysurname>  (optional) | String | The customer’s surname for delivery |
| <deliveryaddress>  (optional) | String  (500 characters) | The customer’s first line of delivery address (without the postcode) if available |
| <deliveryaddress2>  (optional) | String  (500 characters) | The customer’s second line of delivery address (without the postcode) if available |
| <deliverycity>  (optional) | String | The customer’s city for delivery |
| <deliverystate>  (optional) | String | The customer’s state for delivery |
| <deliverypostcode>  (optional) | String  (50 characters) | The customer’s delivery address post code if available |
| <deliverycountry>  (optional) | String | The customer’s country for delivery |
| <deliveryphone>  (optional) | String | The customer’s phone for delivery |
| <cardholder>  (optional) | String  (100 characters) | If available, the name on the credit/debit card. |
| <cardfirstnames>  (optional) | String | The names of the card holder. |
| <cardsurname>  (optional) | String | The surname of the card holder. |
| <cardaddress>  (optional) | String  (500 characters) | The customer’s registered first line of card address (without the postcode) if available |
| <cardaddress2>  (optional) | String  (500 characters) | The customer’s registered second line of card address (without the postcode) if available |
| <cardcity>  (optional) | String | The customer’s registered card address city if available |
| <cardstate>  (optional) | String | The customer’s registered card address state if available |
| <cardpostcode>  (optional) | String  (50 characters) | The customer’s registered card address post code if available |
| <cardcountry>  (optional) | String | The customer’s registered card address country if available |
| <customername>  (optional) | String  (100 characters) | The customer’s name, if available |
| <customeremail>  (optional) | String  (100 characters) | The customer’s e-mail address if available |
| <contactnumber>  (optional) | String  (20 characters) | The customer’s telephone number if available |
| <contactfax>  (optional) | String  (20 characters) | The customer’s fax number if available |
| <vendoremail>  (optional) | String  (100 characters) | The vendor e-mail address passed with this transaction |
| <systemused> | String  (1 character) | Contains **F** for VSPForm, **S** for VSPServer, **D** for VSPDirect or **T** for VSPTerminal. |
| <vpsprotocol> | String | Current version of gateway protocol |
| <callbackurl>  (optional) | String | Callback url to which notification posts are sent |
| <aborted>  (optional) | Datetime | The date/time this transaction was aborted, if applicable |
| <released>  (optional) | Datetime | The date/time this transaction was released, if applicable |
| <refunded>  (optional) | String  (8 characters) | Either **YES** if fully refunded, **NO** if not refunded or **PARTIAL** if partly refunded |
| <repeated>  (optional) | String  (3 characters) | Either **YES** if at least one REPEAT has occurred against this transaction, **NO** if none have |
| <username>  (optional) | String  (20 characters) | If available, the name of the user account in MySagePay who processed this transaction. |
| <basket>  (optional) | String  (7500 characters) | If available, the full basket contents associated with this transaction |
| <basketxml>  (optional) | String | If available, the full basket xml contents associated with this transaction.  Please note this includes the TRIPS data (hotel, cruise, carRental, tourOperator, airline and Diners) |
| <customerxml>  (optional) | String | If available the customer xml data associated with this transaction. |
| <applyavscv2> | Numeric | Value 0 to 3. See the protocols for more information about these values |
| <apply3dsecure> | Numeric | Value 0 to 3. See the protocols for more information about these values |
| <authattempt>  (optional) | Numeric | A value between 1 and 3 to indicate the number of customer attempts to gain and authorisation (if they proceeded that far) |
| <cv2result>  (optional) | String  (12 characters) | The results of the CV2 fraud checks.  Either MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED or PARTIAL |
| <addressresult>  (optional) | String  (12 characters) | The results of the AVS address fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| <postcoderesult>  (optional) | String  (12 characters) | The results of the AVS postcode fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| <threedattempt> | Numeric | The number of customer attempts to gain 3D Secure authentication. |
| <threedresult>  (optional) | String  (12 characters) | The results of the 3D-Secure fraud checks.  Either OK, NOTCHECKED, ATTEMPTONLY, NOTAVAILABLE, NOTAUTHED, INCOMPLETE or ERROR |
| <eci>  (optional) | Numeric | The ECI code sent to bank during authorisation or 3D-Secure transactions |
| <cavv>  (optional) | String  (32 characters) | The encoded CAVV or UCAF value for this secured transaction, if available |
| <t3mscore>  (optional) | Numeric | A value in the range -100 to +100 as a fraud score from the third man, when available. |
| <t3maction>  (optional) | String  (10 characters) | Either **OK, HOLD**, **REJECT** or **NORESULT** as a recommended course of action based on the fraud score. |
| <t3mid>  (optional) | String  (20 characters) | The ID of this transaction in the 3rd Man systems. |
| <fraudscreenrecommendation> (optional) | String (10 characters) | Fraud screening recommendation for the transaction.  Contains either **ACCEPT**, **DENY, CHALLENGE**  or **NOTCHECKED** |
| <fraudcode>  (optional) | String (10 characters) | The fraud provider’s response code. |
| <fraudcodedetail>  (optional) | String (200 characters) | A human-readable explanation of the fraud code |
| <emailmessage>  (optional) | String |  |
| <invalidstatus>  (optional) | String |  |
| <invalidstatusdetail>  (optional) | String |  |
| <invalidrequestsource>  (optional) | String |  |
| <locale> | String |  |
| <xid> | String | An authorisation code that is part of the 3D Secure process. |
| <pprotxid>  (optional) | String | Only for Ppro transactions. |
| <declinecode>  (optional) | String | Return the Decline code for a transaction from the acquiring bank. |
| <vendordata>  (optional) | String | A field that can be used by Vendors to send any information that is relevant to them. |
| <website>  (optional) | String | The website name for the merchants to specify which website the transaction was completed through. Useful to those vendors who have multiple websites. |
| <surcharge>  (optional) | Numeric | The surcharge value charged on this transaction. |

**Sample XML Input 1**

<vspaccess>

<command>getTransactionDetail</command>

<vendor>vendorname</vendor>

<user>username</user>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Input 2**

<vspaccess>

<command>getTransactionDetail</command>

<vendor>vendorname</vendor>

<user>username</user>

<vpstxid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vpstxid>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

  <errorcode>0000</errorcode>

  <timestamp>04/11/2013 11:45:32</timestamp>

  <vpstxid>EE6025C6-7D24-4873-FB92-CD7A66B9494E</vpstxid>

  <vendortxcode>REF20131029-1-838</vendortxcode>

  <transactiontype>Refund</transactiontype>

  <txstateid>29</txstateid>

  <status>Successfully authorised transaction.</status>

  <relatedtransactionid>A5A4A053-2415-3604-35BF-B5EB90D468BB</relatedtransactionid>

  <relatedamount>0.00</relatedamount>

  <description>My Sage Pay REFUND Payment Request by you</description>

  <amount>56.25</amount>

  <currency>EUR</currency>

  <started>29/10/2013 15:59:30.880</started>

  <completed>29/10/2013 15:59:32.410</completed>

  <securitykey>1DCTWJ7X0H</securitykey>

  <clientip>010.000.001.111</clientip>

  <giftaid>NO</giftaid>

  <paymentsystem>SOFORT</paymentsystem>

  <paymentsystemdetails>Debit Card</paymentsystemdetails>

  <authprocessor>PPRO</authprocessor>

  <accounttype>E</accounttype>

  <vpsauthcode></vpsauthcode>

  <bankauthcode></bankauthcode>

  <billingfirstnames>simulate</billingfirstnames>

  <billingsurname>ui</billingsurname>

  <billingaddress>Billing address 1</billingaddress>

  <billingaddress2>Billing address 2</billingaddress2>

  <billingcity>london</billingcity>

  <billingpostcode>ha11tb</billingpostcode>

  <billingcountry>GB</billingcountry>

  <deliveryfirstnames>del first</deliveryfirstnames>

  <deliverysurname>del surname</deliverysurname>

  <deliveryaddress>Delivery address 1</deliveryaddress>

  <deliveryaddress2>Delivery address 2</deliveryaddress2>

  <deliverycity>london</deliverycity>

  <deliverypostcode>ha11tb</deliverypostcode>

  <deliverycountry>GB</deliverycountry>

  <cardholder>simulate ui</cardholder>

  <systemused>S</systemused>

  <vpsprotocol>3.00</vpsprotocol>

  <refunded>NO</refunded>

  <repeated>NO</repeated>

  <username>bindu</username>

  <basket></basket>

  <applyavscv2>0</applyavscv2>

  <apply3dsecure>0</apply3dsecure>

  <authattempt>1</authattempt>

  <cv2result>NOTPROVIDED</cv2result>

  <addressresult>NOTPROVIDED</addressresult>

  <postcoderesult>NOTPROVIDED</postcoderesult>

  <threedresult>NOTCHECKED</threedresult>

  <t3maction>NORESULT</t3maction>

  <emailmessage></emailmessage>

  <locale>en</locale>

  <pprotxid>15005229</pprotxid>

  <surcharge>0.00</surcharge>

</vspaccess>

## Reporting Command - getTransactionIPDetails

This command returns the IP Address and country of origin of the client browser used in the specified transaction. Only information about a transaction associated with the vendor can be returned. Transactions can be specified by either VendorTxCode or VPSTxID.

This is a subset of the getTransactionDetails request, to provide often requested functionality.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

See also: getTransactionSummary, getTransactionList, getBatchDetail

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTransactionIPDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <vendortxcode>  (optional) **OR** | String  (40 characters) | The VendorTxCode of the transaction |
| <vpstxid>  (optional) | String  (40 characters) | The VPSTxID (Transaction ID) of the transaction |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <clientip>  (optional) | String  (15 characters) | The IP address of the customer associated with this transaction (where available) |
| <iplocation>  (optional) | String  (100 characters) | The geolocation of the IP address (where available) |

**Sample XML Input 1**

<vspaccess>

<command>getTransactionIPDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Input 2**

<vspaccess>

<command>getTransactionIPDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<vpstxid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vpstxid>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<clientip>123.123.123.123</clientip>

<iplocation>London</iplocation>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getTransactionList

This command returns a list of all transactions started between the specific dates for the given vendor. You can filter on transaction type, user name and success and failure. By default this command will return all successful transaction information, for all transaction types, between the specified dates.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

NB: The list is limited to 1 month periods to minimise impact on the database. Multiple queries would be needed to return results for longer periods.

NOTE: This command returns a maximum of 50 rows of information. The <startrow> and <endrow> fields show which rows have been returned, with <totalrows> showing the total amount of information available.

See also: getTransactionSummary, getTransactionDetail, getBatchDetail

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTransactionList | The service being requested. |
| <vendor> | String  (16 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <startdate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time from which to begin searching for transactions |
| <enddate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time of the last transaction to be returned. Cannot be more than a month after the start date |
| <relatedtransactionid>  (optional) | String  (40 characters) | The vpstxid of a related transaction |
| <vendortxcode>  (optional) | String  (50 characters) | The vendortxcode of a transaction |
| <systemsused>  (optional) | Node |  |
| └<system>  (optional) | String  (1 character) | F = Form  S = Server  D = Direct  T = Terminal |
| <accounttypes>  (optional) | Node |  |
| └<accounttype>  (optional) | String  (1 character) | E=Ecommerce  M=Mail Order Telephone Order  C=Continuous Authority |
| <username>  (optional) | String  (20 characters) | If specified, only the transactions entered by the specified user through Terminal will be returned. |
| <txtypes>  (optional) | Node | Container for the transaction types being returned |
| └<txtype>  (optional) | String  (20 characters) | The transaction types supported. Contains PAYMENT, REFUND, DEFERRED, REPEAT, REPEATDEFERRED, MANUAL, AUTHENTICATE or AUTHORISE |
| <result>  (optional) | String  (8 characters) | Should contain **SUCCESS** or **FAILURE or INVALID** to indicate if successful or failed or invalid transactions are being requested.  Value **PENDING** will return the pending European (PPRO) transactions. |
| <paymentsystems>  (optional) | node |  |
| └<paymentsystem>  (optional) | String  (7 characters) | Short name of card type |
| <amount>  (optional) | Numeric  (monetary value) | The value of the transaction ( inclusign any surcharge levied) |
| <currency>  (optional) | String  (3 characters) |  |
| <last4digits>  (optional) | Numeric  (4 characters) | Last 4 digits of the card used for the transaction |
| <vpsauthcode>  (optional) | Numeric  (big integer value) | The unique Sage Pay auth code for a successful transaction only |
| <repeated>  (optional) | String  (3 characters) | YES for only repeated transactions, NO for those not yet repeated |
| <released>  (optional) | String  (3 characters) | YES for only released transactions, NO for those not yet released |
| <searchphrase>  (optional) | String  (100 characters) | The phrase to be search on – will search on customer names and address fields |
| <includeaddresses>  (optional) | String  (3 characters) | Either YES or NO |
| <invalids>  (optional) | Numeric  (1 character) | Either  0 **=** no invalids,  1 = include invalids,  2 = invalids only |
| <sorttype>  (optional) | String  (14 Characters) | **Either** ByDate or ByVendorTxCode |
| <sortorder>  (optional) | String  (4 Characters) | Either ASC or DESC |
| <startrow>  (optional) | Numeric |  |
| <endrow>  (optional) | Numeric |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <transactions> | Node | A container for the list of transactions returned |
| └<startrow> | Numeric  (integer value) | The start row of the selected record set |
| └<endrow> | Numeric  (integer value) | The end row of the selected record set |
| └<totalrows> | Numeric  (integer value) | The total rows found based on the criteria provided |
| └<transaction> | Node | A container for the individual transaction |
| └<rownumber> | Numeric  (integer value) | The current row number of the returned results |
| └<vpstxid> | String  (40 characters) | The unique VPSTxId (transaction id) of this transaction |
| └<vendortxcode> | String  (40 characters) | The unique VendorTxCode of this transaction |
| └<result> | String | **SUCCESS** or **FAILURE** or **PENDING** |
| └<transactiontype> | String  (15 characters) | The type of transaction. Values are: Authenticate, Authorise, Deferred, Manual, Payment, PreAuth, Refund, Repeat and RepeatDeferred |
| └<location> | String |  |
| └<systemused> | String  (1 character) | Contains **F** for Form, **S** for Server, **D** for Direct or **T** for Terminal. |
| └<accounttype> | String  (1 character) | Contains **E** for eCommerce, **M** for Mail Order/Telephone Order or **C** for Continuous Authority. |
| └<amount> | Numeric  (monetary value) | The value of the transaction ( including any surcharge levied) |
| └<currency> | String  (3 characters) | The ISO 4217 currency code of this transaction |
| └<started> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date/time that the transaction was registered with Sage Pay |
| └<vpsauthcode>  (optional) | Numeric  (big integer value) | The unique Sage Pay auth code for a successful transaction only |
| └<bankauthcode>  (optional) | String  (20 characters) | The authorisation code from the bank. For PayPal transactions this is the PayPal transaction ID |
| └<cv2result>  (optional) | String  (12 characters) | The results of the CV2 fraud checks.  Either MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED or PARTIAL |
| └<addressresult>  (optional) | String  (12 characters) | The results of the AVS address fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| └<postcoderesult>  (optional) | String  (12 characters) | The results of the AVS postcode fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| └<threedresult>  (optional) | String  (12 characters) | The results of the 3D-Secure fraud checks.  Either OK, NOTCHECKED, ATTEMPTONLY, NOTAVAILABLE, NOTAUTHED, INCOMPLETE or ERROR |
| └<cardholder>  (optional) | String  (100 characters) | If available, the name on the credit/debit card. |
| └<batchid>  (optional) | Numeric  (integer value) | If the transaction has been settled with the bank, this is the ID of that batch |
| └<aborted>  (optional) | Datetime | The date/time this transaction was aborted, if applicable |
| └<released>  (optional) | Datetime | The date/time this transaction was released, if applicable |
| └<refunded>  (optional) | String  (8 characters) | Either **YES** if fully refunded, **NO** if not refunded or **PARTIAL** if partly refunded |
| └<repeated>  (optional) | String  (3 characters) | Either **YES** if at least one REPEAT has occurred against this transaction, **NO** if none have |
| └<username>  (optional) | String  (20 characters) | If available, the name of the user account in My Sage Pay who processed this transaction. |
| └<t3mscore>  (optional) | Numeric | A value in the range -100 to +100 as a fraud score from the third man, when available. |
| └<t3maction>  (optional) | String  (10 characters) | Either **OK, HOLD**, **REJECT** or **NORESULT** as a recommended course of action based on the fraud score. |
| └<t3mid>  (optional) | String  (20 characters) | The ID of this transaction in the 3rd Man systems. |
| └<fraudscreenrecommendation>  (optional) | String (10 characters) | Fraud screening recommendation for the transaction.  Contains either **ACCEPT**, **DENY, CHALLENGE** or **NOTCHECKED** |
| └<fraudcode>  (optional) | String (10 characters) | The fraud provider’s response code. |
| └<fraudcodedetail>  (optional) | String (200 characters) | A human-readable explanation of the fraud code |
| └<paymentsystem>  (optional) | String  (20 characters) | The card type used for the transaction |
| └<last4digits>  (optional) | String  (4 characters) | The last 4 digits of the card number, if available |
| └<statusdetail>  (optional) | String  (300 characters) | the status detail of the transaction |
| └<authprocessor>  (optional) | String  (50 characters) | The name of the acquirer to whom this transaction’s batch was sent |
| └<pprotxid>  (optional) | String | Ppro transaction id if it is a ppro transaction. |
| └<website>  (optional) | String | The website name for the merchants to specify which website the transaction was completed through. Useful to those vendors who have multiple websites. |

**Sample XML Input**

<vspaccess>

<command>getTransactionList</command>

<vendor>onlinecheese</vendor>

<user>barry</user>

<startdate>01/01/2010 00:00:00</startdate>

<enddate>02/01/2010 00:00:00</enddate>

<relatedtransactionid></relatedtransactionid>

<vendortxcode>transaction123</vendortxcode>

<systemsused>

<system>F</system>

<system>S</system>

</systemsused>

<accounttypes>

<accounttype>E</accounttype>

</accounttypes>

<username>Barry</username>

<txtypes>

<txtype>PAYMENT</txtype>

<txtype>REFUND</txtype>

</txtypes>

<result>success</result>

<paymentsystems>

<paymentsystem>VISA</paymentsystem>

<paymentsystem>MAESTRO</paymentsystem>

</paymentsystems>

<amount>10.00</amount>

<currency>GBP</currency>

<last4digits>1234</last4digits>

<vpsauthcode>123456789</vpsauthcode>

<repeated>YES</repeated>

<released>YES</released>

<searchphrase>John Smith</searchphrase>

<includeaddresses>NO</includeaddresses>

<sorttype>ByDate</sorttype>

<sortorder>Desc</sortorder>

<startrow>1</startrow>

<endrow>50</endrow>

<signature>799B11DFF4275AEE76531AEC625FADE0</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<transactions>

<transaction>

<rownumber>1</rownumber>

<vpstxid>69042919-9710-06D1-0ED8-0AC146F9661F</vpstxid>

<vendortxcode>REPprotx.200909023502-961</vendortxcode>

<result>SUCCESS</result>

<transactiontype>Repeat</transactiontype>

<location>MAIN</location>

<systemused>S</systemused>

<accounttype>C</accounttype>

<amount>123.00</amount>

<currency>GBP</currency>

<started>09/09/2010 14:35:29.127</started>

<vpsauthcode>552</vpsauthcode>

<bankauthcode>999778</bankauthcode>

<cv2result>NOTPROVIDED</cv2result>

<addressresult>NOTPROVIDED</addressresult>

<postcoderesult>NOTPROVIDED</postcoderesult>

<threedresult>NOTCHECKED</threedresult>

<cardholder>blah blah</cardholder>

<batchid />

<aborted />

<released />

<refunded>NO</refunded>

<repeated>NO</repeated>

<username>rob</username>

<t3mscore />

<t3maction>NORESULT</t3maction>

<t3mid />

<fraudscreenrecommendation>ACCEPT</fraudscreenrecommendation>

<fraudcode>0600</fraudcode>

<fraudcodedetail>Card found in stolen card database</fraudcodedetail>

<paymentsystem>MasterCard</paymentsystem>

<last4digits>0001</last4digits>

<statusdetail />

<authprocessor>HSBC</authprocessor>

</transaction>

<startrow>1</startrow>

<endrow>50</endrow>

<totalrows>1</totalrows>

</transactions>

<timestamp>20/10/2010 17:42:56</timestamp>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<startrow>1</startrow>

<endrow>50</endrow>

<totalrows>100</totalrows>

<transactions>

<transaction>

<rownumber>1</rownumber>

<vpstxid>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</vpstxid>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<result>SUCCESS</result>

<transactiontype>Repeat</transactiontype>

<systemused>S</systemused>

<accounttype>C</accounttype>

<amount>123.00</amount>

<currency>GBP</currency>

<started>09/09/2010 14:35:29.127</started>

<vpsauthcode>123</vpsauthcode>

<bankauthcode>999999</bankauthcode>

<cv2result>NOTPROVIDED</cv2result>

<addressresult>NOTPROVIDED</addressresult>

<postcoderesult>NOTPROVIDED</postcoderesult>

<threedresult>NOTCHECKED</threedresult>

<cardholder>John</cardholder>

<batchid/>

<aborted/>

<released/>

<refunded>NO</refunded>

<repeated>NO</repeated>

<username>username2</username>

<t3mscore/>

<t3maction>NORESULT</t3maction>

<t3mid/>

<paymentsystem>MasterCard</paymentsystem>

<last4digits>0001</last4digits>

<statusdetail/>

<authprocessor>HSBC</authprocessor>

</transaction>

</transactions>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getTransactionSummary

This command returns a summary report of transactions started between the specific dates for the given vendor (equivalent to the transaction summary reports in My Sage Pay). You can filter on transaction type, user and success and failure. By default this command will return a summary of all successful transactions between the specified dates.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

NB: The list is limited to 1 month periods to minimise impact on the database. Multiple queries would be needed to return results for longer periods. Where there are 50,000 or more records are found, error code 0235 will be returned.

See also: getTransactionList, getTransactionDetail, getBatchDetail

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTransactionSummary | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <startdate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time from which to begin searching for transactions |
| <enddate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time of the last transaction to be returned. Cannot be more than a month after the start date. |
| <result> | String  (8 characters) | Should contain **SUCCESS** or **FAILURE** to indicate if successful or failed transactions are being requested.  If you give **PENDING** it will return European (Ppro) transactions that are pending. |
| <username>  (optional) | String  (20 characters) | If specified, only the transactions entered by the specified user through Terminal will be returned. |
| <txtypes>  (optional) | Node | Container for the transaction types being returned |
| └<txtype>  (optional) | String  (20 characters) | The transaction types supported. Contains **PAYMENT**, **REFUND**, **DEFERRED**, **REPEAT**, **REPEATDEFERRED**, **MANUAL**, **AUTHENTICATE**, **AUTHORISE**, **REPEATDEFERRED** |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <users> | Node | A container for the list of summary records by user |
| └<user> | Node | A container for the individual user records |
| └<username> | String  (20 characters) | The user who took this group of transactions |
| └<txtypes> | Node | Container for the transaction types being returned |
| └<txtype> | Node | Container for the transaction type being returned |
| └<transactiontype> | String  (20 characters) | The type of transaction. Contains **PAYMENT**, **REFUND**, **DEFERRED**, **REPEAT**, **REPEATDEFERRED**, **MANUAL**, **AUTHENTICATE**, **AUTHORISE**, **REPEATDEFERRED** |
| └<currencies> | Node | A container for the currency list |
| └<currency> | String  (3 characters) | Contains the ISO 4217 3-letter currency code of a currency for the transaction totals. |
| └<totalnumber> | Numeric  (integer) | The number of transactions for this user, transaction type and currency |
| └<totalvalue> | Numeric  (monetary value) | The value of the transactions for this user, transaction type and currency |

**Sample XML Input**

<vspaccess>

<command>getTransactionSummary</command>

<vendor>vendorname</vendor>

<user>username</user>

<startdate>01/01/2010 00:00:00</startdate>

<enddate>02/01/2010 00:00:00</enddate>

<result>SUCCESS</result>

<username>username2</username>

<txtypes>

<txtype>PAYMENT</txtype>

<txtype>REFUND</txtype>

</txtypes>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<users>

<user>

<username>username2</username>

<txtypes>

<txtype>

<transactiontype>PAYMENT</transactiontype>

<currencies>

<currency>GBP</currency>

<totalnumber>1</totalnumber>

<totalvalue>10.00</totalvalue>

</currencies>

</txtype>

<txtype>

<transactiontype>REFUND</transactiontype>

<currencies>

<currency>GBP</currency>

<totalnumber>1</totalnumber>

<totalvalue>10.00</totalvalue>

</currencies>

</txtype>

<txtypes>

</user>

</users>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getTransactionTypes

This command returns a list of all transaction types supported by the vendor account.

 This command doesn’t require the user to have administrator rights.

NB: There is no equivalent setTransactionTypes function. Additional transaction types must be requested from the Sage Pay support teams by emailing [support@sagepay.com](mailto:support@sagepay.com), including your vendor name and the transaction types you wish to have added.

See also: getURLs

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTransactionTypes | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <txtypes> | Node | Container for the transaction types being returned |
| └<txtype> | String  (20 characters) | The transaction types supported. Contains **PAYMENT**, **REFUND**, **DEFERRED**, **REPEAT**, **REPEATDEFERRED**, **MANUAL**, **AUTHENTICATE**, **AUTHORISE**, **REPEATDEFERRED** |

**Sample Input XML:**

<vspaccess>

<command>getTransactionTypes</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample Output XML:**

<vspaccess>

<errorcode>0000</errorcode>

<txtypes>

<txtype>PAYMENT</txtype>

<txtype>REFUND</txtype>

</txtypes>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getRelatedTransactions

This command returns a list of all the related transactions of a specific transaction, and which started after a specific date for the given vendor.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

NOTE: This command returns a maximum of 50 rows of information. The <startrow> and <endrow> fields show which rows have been returned, with <totalrows> showing the total amount of information available.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getRelatedTransactions | The service being requested. |
| <vendor> | String  (16 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <vpstxid> | String  (40 characters) | The VPSTxID (transactionid) of the transaction |
| <startdate> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date and time from which to begin searching for transactions |
| <username>  (optional) | String  (20 characters) | If specified, only the transactions entered by the specified user through Terminal will be returned. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <transactions> | Node | A container for the list of transactions returned |
| └<startrow> | Numeric  (integer value) | The start row of the selected record set |
| └<endrow> | Numeric  (integer value) | The end row of the selected record set |
| └<totalrows> | Numeric  (integer value) | The total rows found based on the criteria provided |
| └<transaction> | Node | A container for the individual transaction |
| └<rownumber> | Numeric  (integer value) | The current row number of the returned results |
| └<vpstxid> | String  (40 characters) | The unique VPSTxId (transaction id) of this transaction |
| └<vendortxcode> | String  (40 characters) | The unique VendorTxCode of this transaction |
| └<transactiontype> | String  (15 characters) | The type of transaction. Values are: Authenticate, Authorise, Deferred, Manual, Payment, PreAuth, Refund, Repeat and RepeatDeferred |
| └<systemused> | String  (1 character) | Contains **F** for Form, **S** for Server, **D** for Direct or **T** for Terminal. |
| └<accounttype> | String  (1 character) | Contains **E** for eCommerce, **M** for Mail Order/Telephone Order or **C** for Continuous Authority. |
| └<amount> | Numeric  (monetary value) | The value of the transaction ( including any surcharge levied) |
| └<currency> | String  (3 characters) | The ISO 4217 currency code of this transaction |
| └<started> | Datetime  (dd/mm/yyyy hh:mm:ss) | The date/time that the transaction was registered with Sage Pay |
| └<vpsauthcode>  (optional) | Numeric  (big integer value) | The unique Sage Pay auth code for a successful transaction only |
| └<bankauthcode>  (optional) | String  (20 characters) | The authorisation code from the bank. For PayPal transactions this is the PayPal transaction ID |
| └<cv2result>  (optional) | String  (12 characters) | The results of the CV2 fraud checks.  Either MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED or PARTIAL |
| └<addressresult>  (optional) | String  (12 characters) | The results of the AVS address fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| └<postcoderesult>  (optional) | String  (12 characters) | The results of the AVS postcode fraud checks. Either **MATCHED, NOTCHECKED, NOTPROVIDED, NOTMATCHED** or **PARTIAL** |
| └<threedresult>  (optional) | String  (12 characters) | The results of the 3D-Secure fraud checks.  Either OK, NOTCHECKED, ATTEMPTONLY, NOTAVAILABLE, NOTAUTHED, INCOMPLETE or ERROR |
| └<cardholder>  (optional) | String  (100 characters) | If available, the name on the credit/debit card. |
| └<batchid>  (optional) | Numeric  (integer value) | If the transaction has been settled with the bank, this is the ID of that batch |
| └<aborted>  (optional) | Datetime | The date/time this transaction was aborted, if applicable |
| └<released>  (optional) | Datetime | The date/time this transaction was released, if applicable |
| └<refunded>  (optional) | String  (8 characters) | Either **YES** if fully refunded, **NO** if not refunded or **PARTIAL** if partly refunded |
| └<repeated>  (optional) | String  (3 characters) | Either **YES** if at least one REPEAT has occurred against this transaction, **NO** if none have |
| └<username>  (optional) | String  (20 characters) | If available, the name of the user account in MySagePay who processed this transaction. |
| └<t3mscore>  (optional) | Numeric | A value in the range -100 to +100 as a fraud score from the third man, when available. |
| └<t3maction>  (optional) | String  (10 characters) | Either **OK, HOLD,** **REJECT or NORESULT**  as a recommended course of action based on the fraud score. |
| └<t3mid>  (optional) | String  (20 characters) | The ID of this transaction in the 3rd Man systems. |
| └<fraudscreenrecommendation>  (optional) | String (10 characters) | Fraud screening recommendation for the transaction.  Contains either **ACCEPT**, **DENY**, **CHALLENGE** or **NOTCHECKED** |
| └<fraudcode>  (optional) | String (10 characters) | The fraud provider’s response code. |
| └<fraudcodedetail>  (optional) | String (200 characters) | A human-readable explanation of the fraud code |
| └<billingname>  (optional) | String  (100 characters) | If available, the name used for the billing address. |
| └<billingaddress>  (optional) | String  (500 characters) | If available, the address and postcode used for the billing address. |
| └<deliveryname>  (optional) | String  (100 characters) | If available, the name used for the delivery address. |
| └<deliveryaddress>  (optional) | String  (500 characters) | If available, the address and postcode used for the delivery address. |
| └<cardname>  (optional) | String  (100 characters) | If available, the name on the credit/debit card. |
| └<cardaddress>  (optional) | String  (500 characters) | If available, the address of the credit/debit card holder. |
| └<statusdetail>  (optional) | String  (300 characters) | the status detail of the transaction |

**Sample XML Input**

<vspaccess>

<command>getRelatedTransactions</command>

<vendor>onlinecheese</vendor>

<user>barry</user>

<vpstxid>ABCEC4B5-A74F-40DB-89AA-6CEEC335D3FB</vpstxid>

<startdate>01/01/2010 00:00:00</startdate>

<signature>799B11DFF4275AEE76531AEC625FADE0</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<transactions>

<transaction>

<rownumber>1</rownumber>

<vpstxid>ABCB1FC5-383A-E7C4-3307-000307BEB47D</vpstxid>

<vendortxcode>My-d2cd51ea-d010-442a-b195-960190ec3fca</vendortxcode>

<result>SUCCESS</result>

<transactiontype>Authorise</transactiontype>

<location>MAIN</location>

<systemused>S</systemused>

<accounttype>E</accounttype>

<amount>36.50</amount>

<currency>GBP</currency>

<started>24/05/2010 19:37:46.567</started>

<vpsauthcode>269992888</vpsauthcode>

<bankauthcode>438888</bankauthcode>

<cv2result>MATCHED</cv2result>

<addressresult>MATCHED</addressresult>

<postcoderesult>MATCHED</postcoderesult>

<threedresult>ERROR</threedresult>

<cardholder>Nicolas Nicholls</cardholder>

<batchid>13833</batchid>

<aborted />

<released />

<refunded>NO</refunded>

<repeated>NO</repeated>

<username />

<t3mscore>-12</t3mscore>

<t3maction>OK</t3maction>

<t3mid>559527888</t3mid>

<fraudscreenrecommendation>DENY</fraudscreenrecommendation>

<fraudcode>0600</fraudcode>

<fraudcodedetail>Card found in stolen card database</fraudcodedetail>

<paymentsystem>MasterCard</paymentsystem>

<last4digits>0001</last4digits>

<statusdetail />

<authprocessor>NatWest Streamline</authprocessor>

</transaction>

<startrow>1</startrow>

<endrow>50</endrow>

<totalrows>1</totalrows>

</transactions>

</transactions>

</vspaccess>

## Reporting Command - getGiftAidReport

This command returns a list of Gift Aid transactions and the cardholders' names and addresses.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned. Where there are 50,000 or more records are found, error code 0235 will be returned.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getGiftAid | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <startdate> | String  (23 characters) | The date at which to begin searching for Gift Aid transactions. |
| <enddate> | String  (23 characters) | The date at which to stop searching for Gift Aid transactions |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <giftaidtransactions> | node | Contains the transaction nodes for transactions that had Gift Aid activated. |
| L<transaction> | Node | Details of a given transaction. |
| L<started> | Date  (dd/mm/yyyy hh:mm:ss) | The date the transaction was processed. |
| L<vendortxcode> | String  (50 characters) | The vendor Chosen unique reference for this transaction. |
| L<amount> | Numerical | The value of the transaction. ( including any surcharge levied) |
| L<currency> | String  (3 characters) | Contains the ISO 4217 3-letter currency code of a supported currency. Under which the transaction was processed. |
| L<cardname> | String  (100 characters) | The name of the card holder from the card on which the transaction was processed. |
| L<cardaddress> | String (500 characters) | The address of the card holder. |

**Sample Input XML:**

<vspaccess>

<command>getGiftAid</command>

<vendor>vendorname</vendor>

<user>username</user>

<startdate>07-12-2009 00:00:00.000</startdate>

<enddate>07-01-2010 00:00:00.000</enddate>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample Response XML:**

<vspaccess>

<errorcode>0000</errorcode>

<giftaidtransactions>

<transaction>

<started>01/03/2012 09:26:00</started>

<vendortxcode > MyTransactionCode\_12346</vendortxcode>

<amount>29.86</amount>

<currency>GBP</currency>

<cardname>C H Older</cardname>

<cardaddress>41 Cholder Road, London, AB12 3CD</cardaddress>

</transaction>

<transaction>

<started>01/03/2012 10:26:00</started>

<vendortxcode>MyTransactionCode\_12345</vendortxcode>

<amount>10.25</amount>

<currency>GBP</currency>

<cardname>C H Older</cardname>

<cardaddress>14 Cholder Road, London, AB12 3CD</cardaddress>

</transaction>

</giftaidtransactions>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspacceess>

## Reporting Command - getAllCountries

This command returns a list of all country codes.

 This command doesn’t require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getAllCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <countries> | Node | A container for the countries list. |
| └<country> | Node | A container for the country details. |
| └<countrycode> | String  (2 characters) | Contains the ISO 2-letter country code of a country. |
| └<countryname> | String | The full name of the country. |

**Sample XML Input**

<vspaccess>

<command>getAllCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<countries>

<country>

<countrycode>AF</countrycode>

<countryname>Afghanistan</countryname>

</country>

<country>

<countrycode>AL</countrycode>

<countryname>Albania</countryname>

</country>

</countries>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getBlockedCardRanges

This command returns all card BIN ranges currently in the blocked card range list on the vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getBlockedCardRanges | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <blockedcardranges> | node | Node containing blocked card range elements |
| └<blockedcardrange> | Numeric  (9 characters) | The blocked card BIN range |

**Sample XML Input**

<vspaccess>

<command>getBlockedCardRanges</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<blockedcardranges>

<blockedcardrange>675967511</blockedcardrange>

</blockedcardranges>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getBlockedCountries

This command returns all countries currently in the blocked countries list on the vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getBlockedCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <blockedcountries> | node | Node containing blocked countries |
| └<countryrow> | node | Node containing country row elements |
| └<countrycode> | String  (2 characters) | ISO3166 2 character country code |
| └< country> | String  (100 characters) | The country name |

**Sample XML Input**

<vspaccess>

<command>getBlockedCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<blockedcountries>

<countryrow>

<countrycode>ZW</countrycode>

<country>Zimbabwe</country>

</countryrow>

</blockedcountries>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getBlockedIPs

The getBlockedIPs command gets a list of the currently blocked IPs.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getBlockedIPs | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <ipaddresses> | Node | Node containing the IP address nodes |
| └<ipaddress> | Node | Node containing the IP address and subnet mask values |
| └<address> | String  (15 characters) | The IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |
| └<mask> | String  (15 characters) | The subnet mask for the IP address to be blocked in the format 123.123.123.123 – each number block must be 3 digits so zero padding may need to apply (i.e. 2.5.60.70 would be 002.005.060.070) |

**Sample XML Input**

<vspaccess>

<command>getBlockedIPs</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<ipaddresses>

<ipaddress>

<address>123.123.123.123</address>

<mask>255.255.255.255</mask>

</ipaddress>

</ipaddresses>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getBlockedIssuingCountries

This command returns a list of the card issuing countries which will be prevented from processing transactions against this vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getBlockedIssuingCountries | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <blockedissuingcountries> | Node | The containing node of blocked issuing countries. |
| L<issuingcountry> | Node | The containing node of a single issuing country. |
| L<countrycode> | String  (2 Characters) | ISO-2 character country code. |
| L<country> | String  (100 characters) | The country name |

**Sample XML Input**

<vspaccess>

<command>getBlockedIssuingCountries</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<blockedissuingcountries>

<issuingcountry>

<countrycode>AT</countrycode>

<country>Austria</country>

</issuingcountry>

<issuingcountry>

<countrycode>BB</countrycode>

<country>Barbados</country>

</issuingcountry>

</blockedissuingcountries>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getTokenCount

This command returns the number of tokens the vendor currently has.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTokenCount | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <totalnumber> | Numeric | The total number of tokens held. |

**Sample XML Input**

<vspaccess>

<command>getTokenCount</command>

<vendor>vendorname</vendor>

<user>username</user>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<totalnumber>255</totalnumber>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getTokenDetails

This command returns the details associated to a particular token.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getTokenDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <token> | String  (40 characters) | The token you are requesting the details for |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <token> | String  (40 characters) | The token details have been requested for |
| <cardholder> | String  (50 characters) | The name of the cardholder |
| <last4digits> | String  (4 characters) | Last 4 digits of the card associated with token |
| <paymentsystem> | String  (10 characters) | The payment system associated with token.  Contains either **AMEX, DC** (Diner’s Club)**, DELTA** (Visa Debit)**, JCB, MAESTRO, MC** (MasterCard)**, MC** (MasterCard **Debit), PAYPAL, SWITCH** (UK Maestro)**, UKE** (Visa Electron)or **VISA** (Visa Credit) |
| <corporatecard> | String  (3 characters) | Will contain either **YES** or **NO** to indicate whether or not the card is a corporate card |
| <iscredit> | String  (3 characters) | Will contain either **YES** or **NO** to indicate whether or not the card is a credit card |
| <expirydate> | String  (4 characters) | The expiry date of card associated with token |
| <uses> | Numeric | No of attempts association with this token |

**Sample XML Input**

<vspaccess>

<command>getTokenDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

<token>00AA0A0A-0AA0-00A0-A000-A0000A0AA000</token>

<signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<paymentsystem>Maestro</paymentsystem>

<last4digits>0005</last4digits>

<cardholder>CHOlder</cardholder>

<expirydate />

<corporatecard>NO</corporatecard>

<iscredit>NO</iscredit>

<uses>2</uses>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getT3MDetail

This command returns the 3rd Man fraud screening details for a particular transaction.

 This command doesn’t require the user to have administrator rights but only those transactions that can be viewed by the user account will be returned.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getT3MDetail | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <t3mtxid> | String  (40 characters) | The unique Transaction ID of this transaction |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <t3mresults> | Node | Container for the list of 3rd Man rules. |
| └<rule> | Node | Container for a particular 3rd Man rule. |
| └<description> | Text | Text describing the rule. |
| └<score> | Numeric | The score for the associated rule. |

**Sample XML Input**

<vspaccess>

<command>getT3MDetail</command>

<vendor>vendorname</vendor>

<user>username</user>

<t3mtxid>123456789</t3mtxid> <signature>01234567890ABCDEFGH1A2B3C4D5E6G</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<t3mresults>

<rule>

<description>Phone no is a UK landline</description>

<score>-3</score>

</rule>

<rule>

<description>CV2 Pass</description>

<score>-10</score>

</rule>

<rule>

<description>AVS Address pass</description>

<score>-6</score>

</rule>

<rule>

<description>AVS Postcode pass</description>

<score>-6</score>

</rule>

<rule>

<description>Internet transaction</description>

<score>0</score>

</rule>

<rule>

<description>AVS pass, bill = del, low val</description>

<score>-5</score>

</rule>

<rule>

<description>AVS pass, bill = del, low val</description>

<score>-5</score>

</rule>

<rule>

<description>AVS pass, bill = del, low val</description>

<score>-5</score>

</rule>

<rule>

<description/>

<score/>

</rule>

</t3mresults>

<timestamp>01/03/2012 12:34:56</timestamp>

</vspaccess>

## Reporting Command - getUserDetails

This command returns user information used for password reminders and communications

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getUserDetails | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |
| <vendortxcode> or <vpstxid> | String  (50 characters) | The transaction code or id we want to view fraud screening information on. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <username> | String  (20 characters) | Username looked up |
| <firstname> | String  (70 characters) | First name of user |
| <lastname> | String  (70 characters) | Second name |
| <emailaddress> | String  (255 characters) | Email address for notifications (password reminder etc) |
| <commsoptin> | String  (3 characters) | Either **YES** to opt into to emails from the Sage Pay team, **NO** if not. |

**Sample XML Input**

<vspaccess>

<command>getUserDetails</command>

<vendor>vendorname</vendor>

<user>username</user>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>19/12/2011 11:44:18</timestamp>

<username>jim</username>

<firstname>Jim</firstname>

<lastname>Collins</lastname>

<commsoptin>false</commsoptin>

</getUserDetails>

<vspaccess>

## Reporting Command - getFraudScreenDetail

This command returns the fraud screening details for a particular transaction. The recommendation is returned along with details of the specific fraud rules triggered by the transaction.

This command does not require the user to have administrator rights.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getFraudScreenDetail | The service being requested. |
| <vendor> | String  (16 characters) | The Vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The User name, under the above referenced Vendor account, used to carry out this command. |
| <vendortxcode>  (optional) **OR** | String  (40 characters) | The VendorTxCode of the transaction |
| <vpstxid>  (optional) | String  (40 characters) | The VPSTxID (transactionid) of the transaction |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| <fraudscreenrecommendation> | String (10 characters) | Fraud screening recommendation for the transaction.  If the transaction was fraud screened by ReD this field contains either **ACCEPT**, **DENY, CHALLENGE** or **NOTCHECKED.**  If the transaction was fraud screened by Third Man this field contains either **OK, HOLD**, **REJECT** or **NORESULT.** |
| <fraudid> | String (10 characters) | The unique ID from this transaction on the fraud detail. |
| <fraudcode> | String (10 characters) | The fraud provider’s response code. |
| <fraudcodedetail> | String (200 characters) | A human-readable explanation of the fraud code |
| <fraudprovidername> | String (3  characters) | The provider who performed the fraud screening. This will contain either **ReD** or **T3M** |
| <rules> | String | The rules that cause the transaction to be denied.  This only applies to T3M |

**Sample XML Input**

<vspaccess>

<command>getFraudScreenDetail</command>

<vendor>onlinecheese</vendor>

<user>barry</user>

<vendortxcode>01Jan2010Transaction12345</vendortxcode>

<signature>799B11DFF4275AEE76531AEC625FADE0</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<recommendation>CHALLENGE</recommendation>

< fraudid >123456789</ fraudid >

<fraudcode>0600</fraudcode>

<fraudcodedetail>Card found in stolen card database</fraudcodedetail>

<timestamp>07/10/2010 10:03:39</timestamp>

</vspaccess>

## Reporting Command - getVendorSurcharges

This command will obtain the details of the surcharges stored for the provided vendor account.

 The user name provided must have administrative privileges to run this command.

**Request**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <command> | String  getVendorSurcharges | The service being requested. |
| <vendor> | String  (15 characters) | The vendor name of the Sage Pay account being accessed. |
| <user> | String  (20 characters) | The user name, under the above referenced vendor account, used to carry out this command. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Node/Element** | **Type** | **Description** |
| <errorcode> | 4-digit numeric | 0000 if successful, positive value if an error occurred |
| <error>  (optional) | String  (255 characters) | Human readable error message if errorcode is not 0000. |
| └<vendorsurcharges>  (optional) | node | Existing surcharge entries. |
| └<vendorsurcharge> | Xml container element | Each of the vendorsurcharge comes in one of this element. |
| └<paymentsystem> | String | The name of the payment system that has a surcharge set on the system for it. |
| └<surchargetype> | String | **PERCENTAGE** or **FIXED** |
| └<currency> | String  (3 characters) | Must conform to the ISO 4217 3-letter currency code of a supported currency. |
| └<surcharge> | Decimal | Decimal value to set for the surcharge. |

**Sample XML Input**

<vspaccess>

<command> getVendorSurcharges</command>

<vendor>onlinecheese</vendor>

<user>barry</user>

<signature>799B11DFF4275AEE76531AEC625FADE0</signature>

</vspaccess>

**Sample XML Output**

<vspaccess>

<errorcode>0000</errorcode>

<timestamp>04/02/201315:11:25</timestamp>

<vendorsurcharges>

<vendorsurcharge>

<paymentsystem>MC</paymentsystem>

<currency>GBP</currency>

<surcharge>10.00</surcharge>

<surchargetype>FIXED</surchargetype>

</vendorsurcharge>

<vendorsurcharge>

<paymentsystem>MC</paymentsystem>

<currency>EUR</currency>

<surcharge>2.50</surcharge>

<surchargetype>PERCENTAGE</surchargetype>

</vendorsurcharge>

</vendorsurcharges>

</vspaccess>

# Using Special Characters

In order to use the ‘&’ character in a tag value the tag must be passed with CDATA. For example <password><![CDATA[&egpassword1]]></password>)

This is required because otherwise the ‘&’ character is interpreted as an XML value meaning that error 0001 will be displayed stating invalid XML was passed.

# Full <errorcode> list

The table below lists all <errorcode> values that can be returned from the system, and the meaning of the code (attached in the <error> node)

**Important:** Please note that the wording of the error messages may change at a future date. Therefore, please ensure you reference the error codes themselves, *not* the error messages, when using these commands.

|  |  |
| --- | --- |
| **Code** | **Error Message** |
| 0000 | Success |
| 0001 | The message received was not valid XML. |
| 0002. | Invalid command. XML root node should be vspaccess. |
| 0008 | The specified <vendor> and <user> combination is not valid, or the user may be locked out. |
| 0009 | Error (retrieving/updating or inserting) data. Please email support@sagepay.com with this error. |
| 0010 | Cannot validate the <signature> value. |
| 0011 | You cannot execute the specified command without administrative privileges. |
| 0013 | The command passed in the <command> parameter was not recognised. |
| 0019 | Error parsing the request XML. Check the format of your request message. |
| 0021 | Cannot create the new user. Check that the user does not already exist. |
| 0038 | The start date you have supplied is after the end date. |
| 0039 | The end date is more than one month after the start date. You can only return reports at monthly intervals. |
| 0042 | You must supply one of the following nodes with this command, either <vendortxcode> OR <vpstxid>. |
| 0043 | Unable to find the transaction for the <vendortxcode> or <vpstxid> supplied. |
| 0044 | The user does not have permission to view this transaction. |
| 0045 | You cannot set the administrator password using vspaccess. Please contact support@sagepay.com if you need this reset. |
| 0077 | A valid <homepageurl> value is required. |
| 0078 | A valid <supportemailaddress> value is required. |
| 0099 | Start Row and End Row range too large. The difference between the <startrow> and <endrow> value must be no greater than 50. |
| 0204 | <templatetype> element either not found, or present more than once. |
| 0205 | A valid <templatetype> value is required. |
| 0211 | <startvalue> value must be less than the <endvalue> value. |
| 0235 | More than 50,000 rows returned. Please refine the search. |
| 1115 | <streamlinetnumber> element is only required for a Streamline merchant number. |
| 2097 | The email address provided was not recognised by PayPal. |
| 2098 | The primary admin user cannot be locked. |
| 2099 | You cannot unlock your own user. |
| 2104 | A valid <address> value is required in the format 123.123.123.123. You may need to zero pad your ip address so that each number block contains three digits. |
| 2106 | A valid <mask> value is required in the format 123.123.123.123. You may need to zero pad your subnet mask so that each number block contains three digits. |
| 2128 | The command cannot be run for the vspvendorname provided. |
| 2142 | <vendortxcode> or <vpstxid> element either not found, or present more than once. |
| 2173 | The user must have access to the specified <homepage>. Either provide the user with access to the chosen <homepage> or select another <homepage> which the user has access to. |
| 2180 | You cannot delete your own user. |
| 2188 | <mask> is too restrictive. Only C class subnet masks are allowed for blocking. |
| 2193 | 3D Secure rule not added. Overlaps with an existing rule. |
| 2194 | AVSCV2 rule not added. Overlaps with an existing rule. |
| 2197 | The primary admin user cannot be deleted. |
| 2210 | No running server found |
| 2212 | No successful transaction found within this duration. |
| 2213 | Exceeded the allowed amount of commands to execute. |
| 2214 | IP Address ##### not added. |
| 2216 | <password> can not be less than 8 characters. |
| 2217 | IP Address ##### not deleted. |
| 2219 | Card Range ##### not added. |
| 2220 | Card Range ##### not deleted. |
| 2221 | TerminalId ##### not added. |
| 2222 | <endrow> cannot be before <startrow>, please check. |
| 2223 | <username> and <vendorname> cannot be the same. Please provide different <username> value. |
| 2224 | There is no such username associated with your account. |
| 2225 | <startdate> cannot be after this month. |
| 2226 | <expirydate> cannot be before this month. |
| 2227 | You must provide the first 9 digits of the card number in the <cardbegins> field. |
| 2228 | Cannot complete request. One or more of the specified <terminalids> already exists for the merchant number/currency combination. |
| 2229 | A valid <startvalue> value is required,this must be value between 0 and 100000.00. |
| 2230 | A valid <endvalue> value is required,this must be value between 0 and 100000.00. |
| 2231 | <txtype> values must be either PAYMENT, REFUND, DEFERRED, REPEAT, REPEATDEFERRED, MANUAL, AUTHENTICATE or AUTHORISE. |
| 2232 | A valid <authprocessor> is required. This should be one of the following: Barclays Merchant Services, NatWest Streamline, Lloyds TSB Cardnet, HSBC, American Express, Diners Club, Bank Of Scotland, JCB, Bank Of Ireland, Omnipay, Alliance and Leicester, PayPal, Allied Irish Bank or ChasePaymentTech. |
| 2233 | Provided vendor name does not exist in the system. |
| 2234 | The following currency code(s) are either not valid or not supported:<currency>. Please supply ISO 4217 standard currency codes. |
| 2235 | A valid <amount> value is required,this must be value between 0 and 999999999.99. |
| 2236 | Provided token id does not exist in the system. |
| 2237 | Specified <merchantnumber> already exists. |
| 2238 | Unable to find the transaction for the <vpstxid> supplied. |
| 2241 | <reportingpassword> can not be less than 8 characters. |
| 2242 | Unable to find the transaction for the <t3mtxid> supplied. |
| 2243 | <endrow> cannot be more than 50 if <startrow> is not provided. |
| 2244 | <endrow> cannot be equal to zero. |
| 2245 | Fraud screen detail does not exist for this transaction. |
| 2254 | The given vendorsurcharge cannot be found. |
| 2255 | This Surcharge already exists (paymentsystem and currency combination). |
| 2256 | The <paymentsystem> supplied is not available on any merchant accounts associated with this vendor account. |
| 2257 | The <currency> supplied is not available on any merchant accounts associated with this vendor account. |
| 2258 | Either the currency requested is not available for this payment system, or the supplied combination of currency and paymentsystem is not available for this vendor account. |
| 2260 | Surcharge not allowed on this payment system. |
| 2261 | Negative surcharge value is not allowed. |
| 2262 | Surcharge value for percentage cannot be greater than 100. |
| 2263 | The following language code(s) are either not valid or not supported: <defaultlanguage>. |
| 2264 | Incorrect Fixed value entered. Only zero or positive decimal values, to 2 decimal places. |
| 2265 | Incorrect Fixed value entered. Only zero or positive integer. |
| 2266 | Incorrect Percentage value entered. Only zero or positive decimal value, to 2 decimal places. |
| 2267 | PPro is not enabled for this vendor. |
| 3000 | ##### element present more than once. |
| 3001 | ##### element not found. |
| 3002 | A valid ##### value is required. |
| 3003 | A valid ##### value is required, this should be either ##### or #####. |
| 3004 | You must provide at least one ##### element within ##### node. |
| 3005 | XML does not match the protocol. ##### tag either not required for this command or XML structure is incorrect. |
| 3006 | ##### must be a numeric value. |
| 3007 | ##### must be a numeric value with up to 2 decimal places. |
| 3008 | ##### supplied is too long. |
| 3009 | ##### is an invalid date format. Please use format dd/MM/yyyy HH:mm:ss. |
| 3010 | Duplicate Key |
| 8888 | Unable to complete request. Please email support@sagepay.com with this error. |
| 8890 | This service is temporarily unavailable due to maintenance. |
| 9074 | <merchantnumber> does not exist. |
| 9999 | Unable to complete request. Please email support@sagepay.com with this error.ctionDetail |