



vtwsclib – vtiger CRM Web Services
Client Library
version 1.3

Table of Contents

Version History	4
About	5
Installation	6
Requirement	6
PHP Client Library Overview	7
Class overview	7
Vtiger_WSClient(url)	7
Boolean doLogin(username, accesskey)	7
Object doListTypes()	9
Object doDescribe(module)	10
Object doCreate(module, valuesmap)	11
Object doRetrieve(record)	12
Object doQuery(query)	13
Object doInvoke(method, params, type)	14
Object lastError()	15
Object toJSON(inputString)	16
String toJSONString(input)	16
Javascript Client Library Overview	17
Class overview	17
Vtiger_WSClient(url)	17
doLogin(username, accesskey, callback)	18
Callback Argument	18
doListTypes(callback)	19
doDescribe(module, callback)	20
doCreate(module, valuesmap, callback)	21
doRetrieve(record, callback)	22
doQuery(query, callback)	23
doInvoke(callback, method, params, type)	24
lastError()	25
Object toJSON(inputString)	26
String toJSONString(input)	26
Python Client Library Overview	27
Class overview	27
Vtiger_WSClient(url)	27
Object doLogin(username, accesskey)	28
Object doListTypes()	29
Object doDescribe(module)	30
Object doCreate(module, valuesmap)	31
Object doRetrieve(record)	32
Object doQuery(query)	33
Object doInvoke(method, params, type)	34
Object lastError()	35

Object toJSON(inputString)	36
String toJSONString(input)	36
vtclient application	37
Login Screen	37
Home Page	38
Browse/List view	39
Configure Columns	40
Searching	41
Detail view	43
Preferences	44
Appendix – I : API Changes	46
doListTypes	46

Version History

Version	Release Date	Highlights
1.3	2009.09.22	* Added doInvoke API to invoke custom webservice functions
1.2	2009.03.28	* Added lastError API to get the last operation error information. * Changed doListTypes API result datastructure to Map instead of List.
1.1	2008.11.06	* Minor bug fix
1.0	2008.10.23	* Provide Basic API wrapper for Webservice support

About

vtwsclib (vtiger CRM Web Services Client Library) provides wrapper classes to use vtiger CRM web services. Using this library you can write new application that integrate with vtiger CRM easily.

The current release of vtwsclib library provides web services client wrappers for PHP, JS and Python languages.

Installation

Unpack the `vtwsclib-x.y.zip` into the folder where your web services client application will reside.

Requirement

You will need to enable the `php_curl` extension when using PHP Client Library.

PHP Client Library Overview

PHP Client wrapper (class `Vtiger_WSCClient`) provides the following functions:

- `doLogin`
- `doListTypes`
- `doDescribe`
- `doCreate`
- `doRetrieve`
- `doQuery`
- `doInvoke`
- `lastError`
- `toJSON`
- `toJSONString`

Class overview

class `Vtiger_WSCClient` is defined in the file `vtwsclib/Vtiger/WSCClient.php`. Include this class in your PHP web services client application code as illustrated below

Vtiger_WSCClient(url)	
Parameters	url – vtiger CRM URL
Return value	Instantiates class object
Example	<pre><?php include_once('vtwsclib/Vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); ?></pre>

Boolean doLogin(username, accesskey)	
Parameters	username – vtiger CRM username accesskey – users unique key (See My Preferences in vtiger CRM as shown here)
Return value	True – on success False – on failure
Details	This function initializes <code>Vtiger_WSCClient</code> with <code>sessionId</code> on successful login which is required for performing other operations.
Example	<pre><?php include_once('vtwsclib/Vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$login = \$client->doLogin('admin', 'KpS9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else echo 'Login Successful';</pre>

?>

vtiger CRM User Preferences

vtiger CRM 5 My Preferences Help About Us Sign Out (admin)

My Home Page Marketing Sales Support Analytics Inventory Tools Settings Quick Create... Search...

My Preferences

Viewing details about the user "Administrator"

[View Audit Trail](#) [Edit](#)

1. User Login & Role

User Name	admin	Admin	on
Password	Change Password	Email	prasad@mail.localhost.com
Status	Active	First Name	
Currency	Dollar : \$	Last Name	Administrator
Default Lead View	Today	Role	CEO
Default Calendar View	This Week		

2. More Information

5. User Advanced Options

Reminder Interval	1 Minute	Access Key	KpS9EjNz16JtPmoe
-------------------	----------	------------	------------------

Object doListTypes()	
Parameters	
Return value	Module information accessible to logged in user.
Details	
Example	<pre><?php include_once('vtwsclib/Vtiger/WSClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSClient(\$url); \$login = \$client->doLogin('admin', 'KpS9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { \$modules = \$client->doListTypes(); foreach(\$modules as \$modulename => \$moduleinfo) { echo "ModuleName: \$modulename\n
"; } } ?></pre>

Object <code>doDescribe(module)</code>	
Parameters	module – module name
Return value	Details of the module as Map
Details	Information about fields of the module, permission to create, delete, update records of the module can be obtained.
Example	<pre> <?php include_once('vtwscLib/Vtiger/wSClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSClient(\$url); \$login = \$client->doLogin('admin', 'kps9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { \$modules = \$client->doListTypes(); foreach(\$modules as \$modulename => \$moduleinfo) { \$describe = \$client->doDescribe(\$module); \$cancreate = \$describe[createable]; \$canupdate = \$describe[updateable]; \$candelete = \$describe[deleteable]; \$canread = \$describe[retrieveable]; \$fields = \$describe[fields]; } } ?> </pre>

Object <code>doCreate(module, valuesmap)</code>	
Parameters	module – module name valuesmap – field name and field value pairs
Return value	On successful creation New record information will be returned. Creation may fail if mandatory field name and value pair is not set.
Details	Creating new records in the module can be done using this method.
Example	<pre> <?php include_once('vtwsclib/vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$login = \$client->doLogin('admin', 'KpS9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { \$module = 'Leads'; \$record = \$client->doCreate(\$module, Array('lastname'=>'CRMLead', 'company'=>'vtiger')); if(\$record) { \$recordid = \$client->getRecordId(\$record['id']); } } ?> </pre>
NOTE:	The server returns record['id'] in the format <moduleid>'x'<recordid>. Use method getRecordId method can be used to retrieve only the record id part from the returned record id from the server.

Object <code>doRetrieve(record)</code>	
Parameters	record – identifier having the format <code><moduleid>'x'<recordid></code>
Return value	If specified module record is found its information is returned otherwise False. Creation may fail if mandatory field name and value pair is not set.
Details	Retrieve information of existing record of the module.
Example	<pre> <?php include_once('vtwscLib/Vtiger/wSClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSClient(\$url); \$login = \$client->doLogin('admin', 'Kps9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { // Retrieve Contacts record with id 110 \$record = '4x110'; \$recordInfo = \$client->doRetrieve(\$record); if(\$recordInfo) { \$lastname = \$recordInfo['lastname']; } } ?> </pre>

Object doQuery(query)	
Parameters	query – SELECT query to use for fetching the records.
Return value	
Details	Webservices provides custom subset of SQL support to work with vtiger CRM.
Example	<pre> <?php include_once('vtwsclib/vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$login = \$client->doLogin('admin', 'KpS9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { // Retrieve Accounts matching name 'vtiger' \$query = "SELECT * FROM Accounts WHERE accountname LIKE 'vtiger'"; \$records = \$client->doQuery(\$query); if(\$records) { \$columns = \$client->getResultColumns(\$records); foreach(\$records as \$record) { // Process record information } } } ?> </pre>
NOTE:	<p>Use getResultColumns API to retrieve column names available in the result records.</p> <p>Please refer the webservice documentation to know more about SQL format and support.</p>

Object <code>doInvoke</code> (method, params, type)	
Parameters	method – Name of the webservice method to invoke params – Parameters to the webservice method type – POST/GET transport type to use to invoke the method
Return value	
Details	This method provides the ability to invoke the custom webservice method that is exposed by REST interface.
Example	<pre> <?php include_once('vtwsclib/vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$login = \$client->doLogin('admin', 'KpS9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { \$response = \$client->doInvoke('custom_wsmethod', array('firstparam' => 'F', 'secondparam' => 'S', 'POST')); \$wasError = \$client->lastError(); if(\$wasError) { var_dump(\$client->lastError()); } else { // Take action on \$response } } ?> </pre>

Object <code>lastError()</code>	
Parameters	
Return value	Map containing (code, message) if the last operation failed or else false
Details	
Example	<pre> <?php include_once('vtwsclib/vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$login = \$client->doLogin('admin', 'Kps9EjNz16JtPmoe'); if(!\$login) echo 'Login Failed'; else { // Retrieve Accounts matching name 'vtiger' \$query = "SELECT * FROM Accounts WHERE WRONGQUERY"; \$records = \$client->doQuery(\$query); \$wasError= \$client->lastError(); if(\$wasError) { echo \$wasError['code'] . ':' . \$wasError['message']; } } ?> </pre>
NOTE	The lastError information is reset after every operation is executed.

Object toJSON (inputString)	
Parameters	inputString – json format input string
Return value	PHP Array or Map or String
Details	Converts the given input JSON format string to PHP Object
Example	<pre><?php include_once('vtwsclib/Vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$phpmap = \$client->toJSON('{ "A" : "B" }'); ?></pre>
String toJSONString (input)	
Parameters	input – PHP Array or Map or String
Return value	JSON string
Details	Converts the given input to JSON format string
Example	<pre><?php include_once('vtwsclib/Vtiger/WSCClient.php'); \$url = 'http://en.vtiger.com/wip'; \$client = new Vtiger_WSCClient(\$url); \$jsonstring = \$client->toJSONString({ 'a'=>'ValueA' }); ?></pre>

Javascript Client Library Overview

The following script includes are required on your page to use web services through Javascript client:

```
<script type='text/javascript' src='vtwsclib/third-party/js/jquery.js'></script>  
<script type='text/javascript' src='vtwsclib/third-party/js/md5.js'></script>  
<script type='text/javascript' src='vtwsclib/Vtiger/WSClient.js'></script>
```

Javascript Client wrapper (Vtiger_WSClient) provides the following functions:

- doLogin
- doListTypes
- doDescribe
- doCreate
- doRetrieve
- doQuery
- doInvoke
- lastError
- toJSON
- toJSONString

Class overview

class Vtiger_WSClient is defined in the file vtwsclib/Vtiger/WSClient.js

Vtiger_WSClient(url)	
Parameters	url - vtiger CRM URL
Return value	Instantiates class object
Example	<pre><script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSClient(url); </script></pre>

doLogin(username, accesskey, callback)	
Parameters	username – vtiger CRM username accesskey – users unique key (See My Preferences in vtiger CRM as shown here) callback – callback argument (See details explained here)
Callback First parameter	True – on success False – on failure
Details	Initializes Vtiger_WSClient with sessionId on successful login which is required for performing other operations. Function referenced via callback will be invoked.
Example	<pre><script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSClient(url); client.doLogin('admin', 'kps9EjNz16JtPmoe', callback); // postLogin function gets a call once request is completed function postLogin(result, args) { if(result) alert('Login was successful'); else alert('Login failed'); } </script></pre>

Callback Argument

Asynchronous calls are made to service using AJAX request by webservice client functions.

Hence callback function should be sent, which will be invoked after the response is obtained from the server.

The callback argument can be any of the following:

- Function Reference
- Callback map having the following format:


```
{
  'function' : <Function Reference>,
  'arguments' : {'arg1' : 'value1', 'arg2' : ['value2']}
}
```

Webservice function will invoke the callback function passing the result as the first parameter. If callback map was sent then the reference sent via 'function' key will be invoked with result as the first parameter and second parameter will be values referenced by 'arguments' key of the callback map.

doListTypes(callback)	
Parameters	callback – callback argument (See details explained here)
Callback First parameter	Module information accessible to logged in user.
Details	
Example	<pre><script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSClient(url); client.doLogin('admin', 'Kps9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else getModules(); } function getModules() { client.doListTypes(postGetModules); // postGetModules gets a call once request is completed } function postGetModules(modules, args) { if(modules) alert(client.toJSONString(modules)); } </script></pre>

doDescribe(module, callback)	
Parameters	module – module name callback – callback argument (See details explained here)
Callback First Parameter	Details of the module as Map
Details	Information about fields of the module, permission to create, delete, update records of the module can be obtained.
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); client.doLogin('admin', 'kps9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else getModuleDetails(); } function getModuleDetails() { var module = 'Leads'; var callback = { 'function' : processModuleDetails, 'arguments': { 'moduleName' : module } }; client.doDescribe(callback); // processModuleDetails gets a call once request is completed } function processModuleDetails(result, args) { var module = args.moduleName; if(result) alert('Module = ' + module + ', Details = ' + client.toJSONString(result)); } </script> </pre>

doCreate(module, valuesmap, callback)	
Parameters	module – module name valuesmap – field name and field value pairs callback – callback argument (See details explained here)
Callback First Parameter	On successful creation New record information will be sent. Creation may fail if mandatory field name and value pair is not set.
Details	Creating new records in the module can be done using this method.
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); client.doLogin('admin', 'KpS9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else createModuleRecord(); } function createModuleRecord() { var module = 'Leads'; var valuesmap = { 'lastname' : 'CRMLLead', 'company':'vtiger' }; client.doCreate(module, valuesmap, afterCreateRecord); // afterCreateRecord gets a call once request is completed } function afterCreateRecord(result, args) { if(result) alert('Record Id = ' + client.getRecordId(result.id)); } </script> </pre>
NOTE:	The server returns record['id'] in the format <moduleid>'x'<recordid>. Use method getRecordId method can be used to retrieve only the record id part from the returned record id from the server.

doRetrieve(record, callback)	
Parameters	record – identifier having the format <moduleid>'x'<recordid> callback – callback argument (See details explained here)
Callback First Parameter	If specified module record is found its information is sent, otherwise False Creation may fail if mandatory field name and value pair is not set.
Details	Retrieve information of existing record of the module.
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); client.doLogin('admin', 'KpS9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else getModuleRecord(); } function getModuleRecord() { var record = '4x110'; client.doRetrieve(record, processModuleRecord); // processModuleRecord gets a call once request is completed } function processModuleRecord(result, args) { if(result) alert('Record Id = ' + client.getRecordId(result.id)); } </script> </pre>

doQuery(query, callback)	
Parameters	query – SELECT query to use for fetching the records. callback – callback argument (See details explained here)
Callback First Parameter	Result sent from the server.
Details	Webservices provides custom subset of SQL support to work with vtiger CRM.
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); client.doLogin('admin', 'kps9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else execQuery(); } function execQuery() { var query = "SELECT * FROM Accounts " + "WHERE accountname LIKE '%vtiger%' "; client.doQuery(query, postExecQuery); // postExecQuery gets a call once request is completed } function postExecQuery(result, args) { if(result) { var columns = client.getResultColumns(result); alert('COLUMNS: ' + client.toJSONString(columns)); alert(client.toJSONString(result)); } } </script> </pre>
NOTE:	Use getResultColumns API to retrieve column names available in the result records. Please refer the webservices documentation to know more about SQL format and support.

doInvoke(callback, method, params, type)	
Parameters	callback – callback argument (See details explained here) method – Name of the webservice method to invoke params – Parameters to the webservice method type – POST/GET transport type to use to invoke the method
Callback First Parameter	Result sent from the server.
Details	This method provides the ability to invoke the custom webservice method that is exposed by REST interface.
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); client.doLogin('admin', 'KpS9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else invokeMethod(); } function invokeMethod() { client.doInvoke(postInvoke, 'custom_wsmethod, { 'firstparam' : 'F', 'secondparam' : 'S' }); // postInvoke gets a call once request is completed } function postInvoke(result, args) { if(result == false) { alert(client.lastError()['message']); } else { // Process result } } </script> </pre>

lastError()	
Parameters	
Callback First parameter	Map containing (code, message) if the last operation failed or else false
Details	
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSClient(url); client.doLogin('admin', 'Kps9EjNz16JtPmoe', postLogin); // postLogin function gets a call once request is completed function postLogin(result, args) { if(!result) alert('Login failed'); else execQuery(); } function execQuery() { var query = "SELECT * FROM Accounts WRONGQUERY"; client.doQuery(query, postExecQuery); // postExecQuery gets a call once request is completed } function postExecQuery(result, args) { if(!result) { var wasError = client.lastError(); alert('ERROR CODE: ' + wasError['code'] + ', MESSAGE: ' + wasError['message']); } } </script> </pre>
NOTE	The lastError information is reset after every operation is executed.

Object toJSON(inputString)	
Parameters	inputString – json format input string
Return value	Javascript Array or Map or String
Details	Converts the given input JSON format string to Javascript Object
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); var jsobj = client.toJSON('{ "A" : "B" }'); </script> </pre>
String toJSONString(input)	
Parameters	input – Javascript Array or Map or String
Return value	JSON string
Details	Converts the given input to JSON format string
Example	<pre> <script type='text/javascript'> var url = 'http://en.vtiger.com/wip'; var client = new Vtiger_WSCClient(url); var jsonstring = client.toJSONString({ 'a'=>'ValueA' }); </script> </pre>

Python Client Library Overview

Python Client wrapper (vtiger_WSCliient) provides the following functions:

- doLogin
- doListTypes
- doDescribe
- doCreate
- doRetrieve
- doQuery
- doInvoke
- lastError
- toJSON
- toJSONString

Class overview

class Vtiger_WSCliient is defined in the file vtwsclib/Vtiger/WSCliient.py

Vtiger_WSCliient(url)	
Parameters	url – vtiger CRM URL
Return value	Instantiates class object
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSCliient import * # Create Vtiger webservice client client = Vtiger_WSCliient('http://en.vtiger.com/wip')</pre>

Object <code>doLogin(username, accesskey)</code>	
Parameters	username – vtiger CRM username accesskey – users unique key (See My Preferences in vtiger CRM as shown here)
Return value	True – on success False – on failure
Details	Initializes <code>Vtiger_WSClient</code> with <code>sessionId</code> on successful login which is required for performing other operations. Function referenced via callback will be invoked.
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from wsClient import * # Create vtiger webservice client client = Vtiger_WSClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'KpS9EjNz16JtPmoe') if login: print 'Login Successful' else: print 'Login Failed!'</pre>

Object doListTypes()	
Parameters	
Return value	Module information accessible to logged in user.
Details	
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSCClient import * # Create Vtiger webservice client client = Vtiger_WSCClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'KpS9EjNz16JtPmoe') if not login: print 'Login Failed!' else: result = client.doListTypes() for modulekey in result: print 'Module Name: ', result[modulekey]['name']</pre>

Object <code>doDescribe(module)</code>	
Parameters	module – module name
Return value	Details of the module as Map
Details	Information about fields of the module, permission to create, delete, update records of the module can be obtained.
Example	<pre>import sys # Absolute/Relative path of vtwscLib/Vtiger folder sys.path.append('vtwscLib/Vtiger') from WSCClient import * # Create Vtiger webservice client client = Vtiger_WSCClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'Kps9EjNz16JtPmoe') if not login: print 'Login Failed!' else: moduleInfo = client.doDescribe('Leads') print 'ModuleInfo: ', moduleInfo</pre>

Object <code>doCreate(module, valuesmap)</code>	
Parameters	<p>module – module name</p> <p>valuesmap – field name and field value pairs</p>
Return value	<p>On successful creation New record information will be returned back. Creation may fail if mandatory field name and value pair is not set.</p>
Details	<p>Creating new records in the module can be done using this method.</p>
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSClient import * # Create Vtiger webservice client client = Vtiger_WSClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'Kps9EjNz16JtPmoe') if not login: print 'Login Failed!' else: recordInfo = client.doCreate('Leads', {'lastname' : 'CRMLLead', 'company' : 'vtiger' }) print 'RecordId: ', client.getRecordId(recordInfo['id'])</pre>
NOTE:	<p>The server returns record['id'] in the format <moduleid>'x'<recordid>. Use method getRecordId method can be used to retrieve only the record id part from the returned record id from the server.</p>

Object <code>doRetrieve(record)</code>	
Parameters	record – identifier having the format <code><moduleid>'x'<recordid></code>
Return value	If specified module record is found its information is sent, otherwise False Creation may fail if mandatory field name and value pair is not set.
Details	Retrieve information of existing record of the module.
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSCClient import * # Create Vtiger webservice client client = Vtiger_WSCClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'Kps9EjNz16JtPmoe') if not login: print 'Login Failed!' else: # Fetch contact record with id 110 record = '4x110' recordInfo = client.doRetrieve(record) print 'RecordId: ', client.getRecordId(recordInfo['id'])</pre>

Object <code>doQuery(query)</code>	
Parameters	query – SELECT query to use for fetching the records.
Return value	Result sent from the server.
Details	Webservices provides custom subset of SQL support to work with vtiger CRM.
Example	<pre> import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from wsClient import * # Create Vtiger webservice client client = Vtiger_wsClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'Kps9EjNz16JtPmoe') if not login: print 'Login Failed!' else: # Search for Accounts with name like 'vtiger' query = "SELECT * FROM Accounts WHERE accountname LIKE '%vtiger%'" result = client.doQuery(query) if(result): columns = client.getResultColumns(result) print 'Columns: ', columns print 'Result : ', result </pre>
NOTE:	<p>Use getResultColumns API to retrieve column names available in the result records.</p> <p>Please refer the webservices documentation to know more about SQL format and support.</p>

Object <code>doInvoke</code> (method, params, type)	
Parameters	method – Name of the webservice method to invoke params – Parameters to the webservice method type – POST/GET transport type to use to invoke the method
Return value	Result sent from the server.
Details	This method provides the ability to invoke the custom webservice method that is exposed by REST interface.
Example	<pre> import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSClient import * # Create Vtiger webservice client client = Vtiger_WSClient('http://en.vtiger.com/wip') login = client.doLogin('admin', 'Kps9EjNz16JtPmoe') if not login: print 'Login Failed!' else: # Invoke custom method with parameters result = client.doInvoke('custom_wsmethod', {'firstparam':'f', 'secondparam' : 's'}) if(result): # Process the result else: print 'ERROR: ', client.lastError(); </pre>

Object `lastError()`

Parameters

Return value Map containing (code, message) if the last operation failed or else false

Details

Example

```
import sys
# Absolute/Relative path of vtwsclib/Vtiger folder
sys.path.append('vtwsclib/Vtiger')

from WSCClient import *

# Create Vtiger webservice client
client = Vtiger_WSCClient('http://en.vtiger.com/wip')

login = client.doLogin('admin', 'Kps9EjNz16JtPmoe')

if not login:
    print 'Login Failed!'
else:
    # Search for Accounts with name like 'vtiger'
    query = "SELECT * FROM Accounts WHERE WRONGQUERY"
    result = client.doQuery(query)
    if(result):
        columns = client.getResultColumns(result)
        print 'Columns: ', columns
        print 'Result : ', result
    else:
        errorInfo = client.lastError()
        print 'Error Code: ', errorInfo['code']
        print 'Message : ', errorInfo['message']
```

NOTE:


The `lastError` information is reset after every operation is executed.

Object <code>toJSON(inputString)</code>	
Parameters	inputString – json format input string
Return value	Python Array or Map or String
Details	Converts the given input JSON format string to Python Object
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSCClient import * # Create Vtiger webservice client client = Vtiger_WSCClient('http://en.vtiger.com/wip') pyobj = client.toJSON('{ "A" : "B" }');</pre>
String <code>toJSONString(input)</code>	
Parameters	input – Python Array or Map or String
Return value	JSON string
Details	Converts the given input to JSON format string
Example	<pre>import sys # Absolute/Relative path of vtwsclib/Vtiger folder sys.path.append('vtwsclib/Vtiger') from WSCClient import * # Create Vtiger webservice client client = Vtiger_WSCClient('http://en.vtiger.com/wip') jsonstring = client.toJSONString({ 'a'=>'valueA' });</pre>

vtclient application

vtclient is a sample application written using the PHP client library. It provides a simple vtiger CRM client suitable for mobile devices and other web devices that require simple HTML user interfaces. Its features are demonstrated below:

Login Screen



The screenshot shows a login form for the vtiger CRM Client. The form is titled "vtiger CRM Client" and contains the following fields and controls:

- Username:** A text input field containing the value "admin".
- Access Key:** A text input field where the characters are masked with dots.
- vtiger URL:** A text input field containing the value "http://en.vtiger.com/wip".
- Theme:** A dropdown menu currently showing ".Gray".
- Login:** A button located below the Theme field.

Username	vtiger CRM user
Access Key	Access Key of the user (Check My Preferences in vtiger CRM)
vtiger URL	vtiger CRM Location (like, http://en.vtiger.com/wip)
Theme	Select theme for current login

Home Page

On successful login, the home page displays a list of accessible modules links.



The screenshot shows the vtiger CRM home page. At the top left, it says "vtiger CRM". At the top right, it says "Welcome admin | Logout". In the center, it says "Designed using vtiger CRM 5.1.0 REST API (Webservice)". On the left side, there is a list of modules: Accounts, Calendar, Campaigns, Contacts, Documents, Emails, Faq, HelpDesk, Invoice, Leads, Potentials, PriceBooks, Products, PurchaseOrder, Quotes, SalesOrder, Users, and Vendors. At the bottom, it says "© Copyright - 2008, www.vtiger.com".

Browse/List view

You can click on the module link from the Home page to start browsing the records.



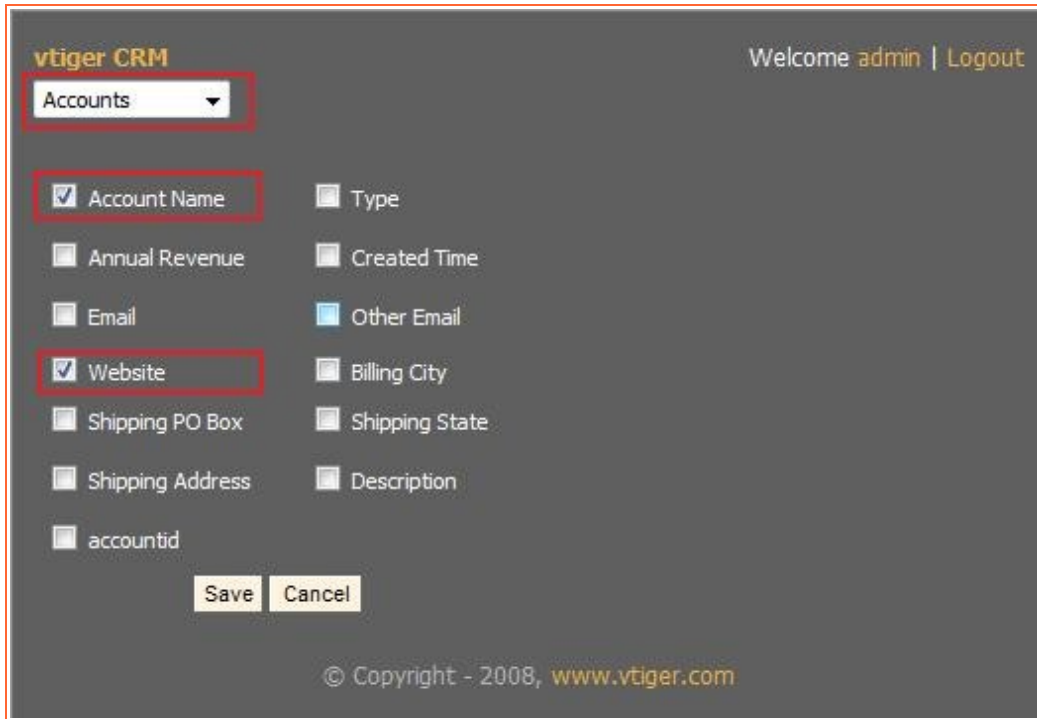
Accountid	Account Name
2	vtiger
4	t3M Invest A/S
6	EDFG Group Limited
8	vtigerCRM Inc
10	X-CEED INC 99
12	demovtiger
14	usable-vtiger
16	gooduivtiger
18	vtigeruser
20	samplevtiger

In the Listview you can perform the following:

- Select other module from the drop down box and continue browsing.
- Click on **Columns** button and configure Listview columns.
- Click on **Search** button and proceed with filtered browsing.

Configure Columns

Click on the **Columns** button to configure module columns.



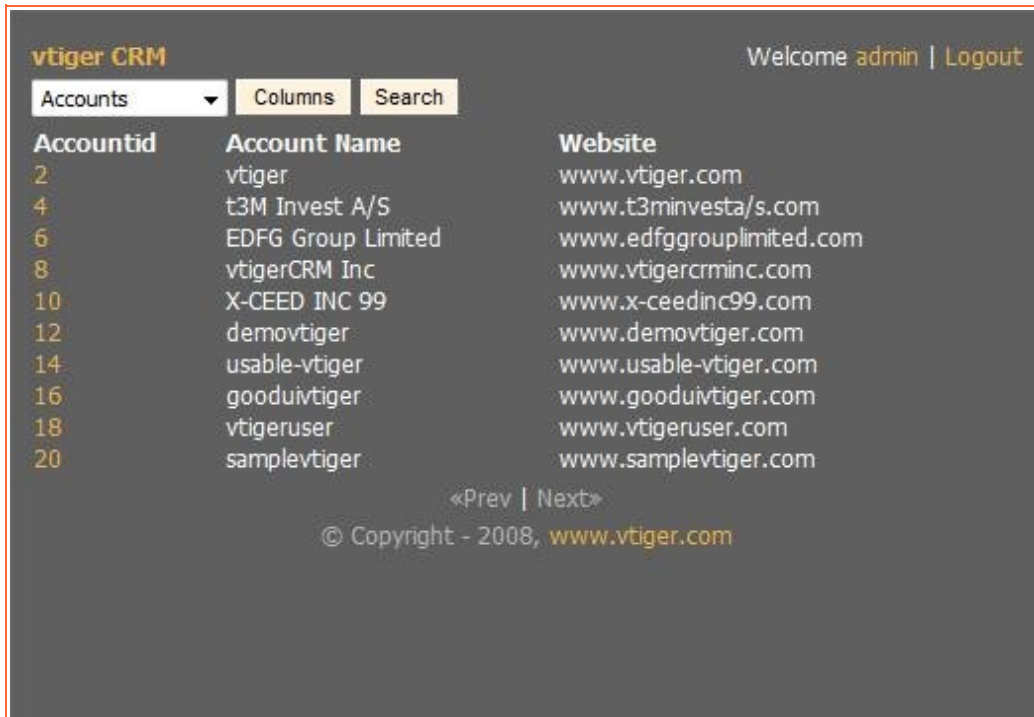
The screenshot shows the 'vtiger CRM' interface for configuring columns for the 'Accounts' module. The 'Accounts' dropdown menu is highlighted with a red box. Below it, a list of columns is shown with checkboxes. 'Account Name' and 'Website' are checked and highlighted with red boxes. Other columns include 'Type', 'Annual Revenue', 'Created Time', 'Email', 'Other Email', 'Billing City', 'Shipping PO Box', 'Shipping State', 'Shipping Address', 'Description', and 'accountid'. At the bottom, there are 'Save' and 'Cancel' buttons. The footer text reads '© Copyright - 2008, www.vtiger.com'.

The columns configured for the module will be considered for:

- Displaying in the Listview
- Performing Search for the module

NOTE: Column configured for the module is persistent till Logout.

Example: Listview after configuring Accounts module.



The screenshot shows the vtiger CRM interface. At the top left, it says "vtiger CRM". At the top right, it says "Welcome admin | Logout". Below the header, there are three buttons: "Accounts" (with a dropdown arrow), "Columns", and "Search". The main content is a table with three columns: "Accountid", "Account Name", and "Website". The table contains 11 rows of data. At the bottom of the table, there are navigation links: "<<Prev | Next>>". At the very bottom, there is a copyright notice: "© Copyright - 2008, www.vtiger.com".

Accountid	Account Name	Website
2	vtiger	www.vtiger.com
4	t3M Invest A/S	www.t3minvesta/s.com
6	EDFG Group Limited	www.edfggrouplimited.com
8	vtigerCRM Inc	www.vtigercrm.com
10	X-CEED INC 99	www.x-ceedinc99.com
12	demovtiger	www.demovtiger.com
14	usable-vtiger	www.usable-vtiger.com
16	gooduivtiger	www.gooduivtiger.com
18	vtigeruser	www.vtigeruser.com
20	samplevtiger	www.samplevtiger.com

Searching

Click on **Search** button to perform module search.



The screenshot shows the vtiger CRM search interface. At the top left, it says "vtiger CRM". At the top right, it says "Welcome admin | Logout". Below the header, there is a "Module" dropdown menu with "Accounts" selected. To the right of the dropdown is a "Columns" button. Below the dropdown is a "Search For" text input field containing "vtiger". Below the input field are two buttons: "Search" and "Cancel". At the bottom, there is a copyright notice: "© Copyright - 2008, www.vtiger.com".

Enter the search input and click **Search** again. The search input is matched against the columns configured for the module and results are displayed.



The screenshot shows the vtiger CRM interface with the 'Accounts' module selected. The search results are displayed in a table with the following columns: Accountid, Account Name, and Website. The 'Columns' and 'Cancel' buttons are highlighted with red boxes.

Accountid	Account Name	Website
2	vtiger	www.vtiger.com
8	vtigerCRM Inc	www.vtigercrm.com
12	demovtiger	www.demovtiger.com
14	usable-vtiger	www.usable-vtiger.com
16	gooduivtiger	www.gooduivtiger.com
18	vtigeruser	www.vtigeruser.com
20	samplevtiger	www.samplevtiger.com

If you select other module, the search is performed on the configured columns of the selected module and results are displayed.



The screenshot shows the vtiger CRM interface with the 'Leads' module selected. The search results are displayed in a table with the following columns: Leadid, First Name, Last Name, Company, and Email. The 'Columns' and 'Cancel' buttons are highlighted with red boxes.

Leadid	First Name	Last Name	Company	Email
32	Mary	Smith	Vtiger	mary_smith@company.com
35	Barbara	Jones	Vtigercrm inc	barbara_jones@company.com
37	Jennifer	Davis	Vtiger	jennifer_davis@company.com
40	Margaret	Moore	Vtigercrm inc	margaret_moore@company.com

The search browsing will be active until Cancel button is clicked. Once you click on Cancel then it returns to normal Browsing for the selected module.

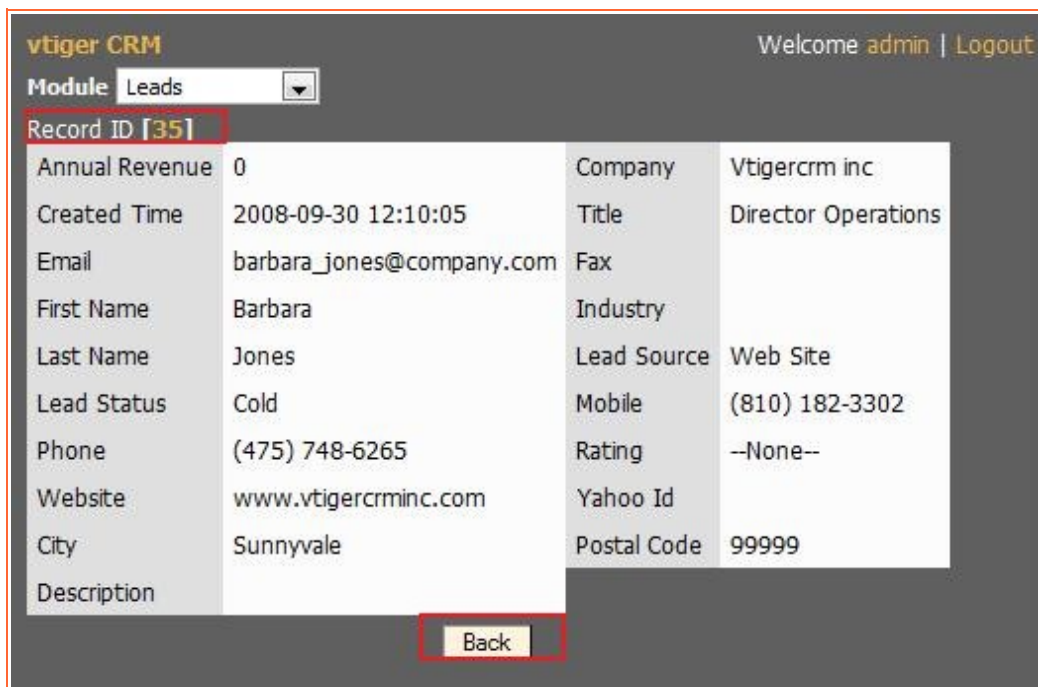


The screenshot shows the vtiger CRM interface with the 'Leads' module selected. A table lists lead records with columns for Leadid, First Name, Last Name, Company, and Email. The 'Leads' dropdown menu is highlighted with a red box. Navigation links for 'Columns' and 'Search' are visible. At the bottom, there are '«Prev | Next»' links and a copyright notice for 2008.

Leadid	First Name	Last Name	Company	Email
32	Mary	Smith	Vtiger	mary_smith@company.com
33	Patricia	Johnson	T3m invest a/s	patricia_johnson@company.com
34	Linda	Williams	Edfg group limited	linda_williams@company.com
35	Barbara	Jones	Vtigercrm inc	barbara_jones@company.com
36	Elizabeth	Brown	X-ceed inc 99	elizabeth_brown@company.com
37	Jennifer	Davis	Vtiger	jennifer_davis@company.com
38	Maria	Miller	T3m invest a/s	maria_miller@company.com
39	Susan	Wilson	Edfg group limited	susan_wilson@company.com
40	Margaret	Moore	Vtigercrm inc	margaret_moore@company.com
41	Dorothy	Taylor	X-ceed inc 99	dorothy_taylor@company.com

Detail view

You can get complete details of the record by clicking the link of record id.



The screenshot shows the detail view for record ID 35 in the 'Leads' module. The 'Record ID [35]' is highlighted with a red box. The details are presented in a table format with a 'Back' button at the bottom, also highlighted with a red box.

Annual Revenue	0	Company	Vtigercrm inc
Created Time	2008-09-30 12:10:05	Title	Director Operations
Email	barbara_jones@company.com	Fax	
First Name	Barbara	Industry	
Last Name	Jones	Lead Source	Web Site
Lead Status	Cold	Mobile	(810) 182-3302
Phone	(475) 748-6265	Rating	--None--
Website	www.vtigercrm.com	Yahoo Id	
City	Sunnyvale	Postal Code	99999
Description			

Click the **Back** button to get back to previous page.

Preferences

You can update other preferences by clicking the username:

Fetch Limit	Number of listview rows displayed on a page.
Theme	Choose a different theme



The screenshot shows the 'vtiger CRM' preferences dialog box. At the top left, it says 'vtiger CRM' and at the top right, 'Welcome admin | Logout'. Below the header is a dropdown menu labeled '-- Select Module --'. The main content area contains two settings: 'Fetch Limit' with a text input field containing the number '5', and 'Theme' with a dropdown menu showing 'Red'. Below these settings are two buttons: 'Save' and 'Cancel'. At the bottom center, there is a copyright notice: '© Copyright - 2008, www.vtiger.com'.

NOTE: Preferences setup is persistent till Logout.

Example: Browsing with Fetch Limit 5 and different theme showing Paging.



The screenshot shows the vtiger CRM interface with a dark red theme. At the top left, it says "vtiger CRM". At the top right, it says "Welcome admin | Logout". Below the header, there are three buttons: "Accounts" (with a dropdown arrow), "Columns", and "Search". The main content area displays a table with two columns: "Accountid" and "Account Name". The table contains five rows of data. At the bottom of the table, there is a pagination control with the text "«Prev | Next»". Below the table, there is a copyright notice: "© Copyright - 2008, www.vtiger.com".

Accountid	Account Name
12	demovtiger
14	usable-vtiger
16	gooduivtiger
18	vtigeruser
20	samplevtiger

«Prev | Next»

© Copyright - 2008, www.vtiger.com

Appendix – I : API Changes

doListTypes

version 1.1: List of module names was returned

```
[ 'Module1', 'Module2', ...]
```

version 1.2: Module information is returned as Map

```
{  
  'Module1' : { 'name' : MODULENAME },  
  'Module2' : { 'name' : MODULENAME }  
}
```