



MINDBODY, INC. Web Service API End User Guide

API Version 0.4

- [MINDBODY, INC. WEB SERVICE API END USER GUIDE.....1](#)**
- [API VERSION 0.4.....1](#)**
- [DOCUMENTATION CHANGE LOG.....4](#)**
- [QUICK START7](#)**
- [REQUEST SOURCE CREDENTIALS.....7](#)
- [REQUEST METHOD PERMISSIONS.....7](#)
- [DEVELOP APPLICATION AND TEST.....8](#)
- [RUNTIME PLATFORM.....9](#)
- [SERVICE AVAILABILITY.....9](#)
- [PREVIEW.MINDBODYONLINE.COM AND CLIENTS.MINDBODYONLINE.COM.....9](#)
- [VERSIONING.....9](#)
- [WSDL.....10](#)
- [API WEBSITE.....11](#)**
- [DEVELOPMENT ASSISTANCE.....12](#)**
- [EXAMPLE MBO WS API CONSUMER APPLICATION.....13](#)**
- [METHOD LIST.....15](#)**
- [APPOINTMENT SERVICE.....15](#)
- [*getAppointment*.....15](#)
- [*getAppointmentByCardID*.....15](#)
- [CLASS SERVICE.....16](#)
- [*getClassesWithClientInfoByCardID*.....16](#)
- [*getClasses*.....16](#)
- [*getClass*.....16](#)
- [*getClassWithClientInfo*.....17](#)
- [CLIENT SERVICE.....17](#)
- [*addArrivalEntry*.....17](#)
- [*addClient*.....18](#)
- [*getClient*.....18](#)
- [*getClientFitRewards*.....19](#)
- [*getClientHasContract*.....19](#)
- [*getClientHasMembership*.....20](#)
- [*getClientTypeList*.....20](#)
- [*getCardIDHistory*.....20](#)
- [*getClientIndexes*.....21](#)
- [*getClientIndexesOnClient*.....21](#)
- [*updateClient*.....22](#)
- [*updateClientIndexOnClient*.....22](#)
- [*updateClientTypeOnClient*.....23](#)
- [*updateLoginNameAndPassword*.....23](#)
- [*updatePassword*.....24](#)
- [*ValidateUserLogin*.....24](#)
- [REGION SERVICE.....25](#)



<i>getLocations</i>	25
SALES SERVICE	25
<i>addSaleSimple</i>	25
<i>addCourseAndPaymentToCart</i>	26
<i>addEnrollmentAndPaymentToCart</i>	26
<i>addItemToCart</i>	26
<i>checkoutShoppingCart</i>	26
<i>getServicePrograms</i>	27
<i>getServices</i>	27
<i>getServicesAndDepositsForCourse</i>	27
<i>getServicesAndDepositsForEnrollment</i>	27
<i>getShoppingCart</i>	28
<i>removeItemFromCart</i>	28
SELECT SERVICES	28
<i>SelectData</i>	28
STAFF SERVICE	29
<i>getStaffList</i>	29
OUTBOUND SCHEMAS	29
<i>AppointmentEvent</i>	30
<i>ClientEvent</i>	30
DATA STRUCTURES	31
<i>CredentialsHeader</i>	31
<i>Result / Result_Bool / Result_Int / Result_Client / Result_Appointment / Result_Appointments /</i>	
<i>Result_StringList / Result_ClientIndexList</i>	31
<i>AppointmentEventInfo</i>	31
<i>ErrorCode</i>	32
<i>PaymentMethod</i>	32
<i>Appointment</i>	32
<i>Class</i>	33
<i>AppointmentStatusDetails</i>	33
<i>Client</i>	34
<i>ClientType</i>	35
<i>Product</i>	35
<i>VisitType</i>	35
<i>TypeGroup</i>	35
<i>Trainer</i>	36
<i>Studio</i>	36
<i>SeriesInfo</i>	36
<i>SeriesType</i>	37
<i>DateTimeFormat</i>	37
<i>ClassVisitStatusDetail</i>	37
<i>Location</i>	37
<i>ClientType</i>	38
<i>ClientIndex</i>	38
<i>ClientIndexValue</i>	38
API USE WITH MULTIPLE STUDIOS	39
STUDIO ID WILD CARD	39
<i>Field Definitions</i>	40
<i>Search by Postal Code</i>	41
<i>Search by RegionID</i>	41
<i>Studio List Search</i>	41
<i>Single Studio Search</i>	41
<i>Class list example</i>	42



<u>Enrollment list example</u>	<u>42</u>
<u>Trainer List Example</u>	<u>42</u>
<u>Substitute Demo</u>	<u>42</u>

Documentation Change Log

July 29, 2009

- Added instructions for owner issued source credentials

March 16, 2009

- Added new service: RegionService
- Added several new methods to SaleService
- Added enrollment methods to ClassService
- Added updateClientWithLoginInformation call to ClientService

February 4th, 2009

- addSale now accepts barcode Ids for services
- Added 'available' boolean to getClasses results

December 18, 2008

- Added ValidateUserLogin method to ClientService
- Added new fields to getClasses results
- All results have now have a Hotwords node for future use

October 28, 2008

- Added updateClient method to ClientService
- Added getStaffList method to StaffService
- Updated Trainer Class

September 25, 2008

- Added SelectData method to SelectService

July 25, 2008

- Added Enrollment list example

May 21, 2008

- Renamed getClientClientIndexes to getClientIndexesOnClient.
- Added getClientTypeList
- Added updateClientTypeOnClient
- Added addArrivalEntry

May 14, 2008

- Added getClientIndexes.
- Added getClientClientIndexes.
- Add updateClientIndexOnClient.
- Added ClientIndex data type.
- Added ClientIndexValue data type.

May 7, 2008

- Split getClientHasMembership into getClientHasMembership and getClientHasContract.

April 29, 2008

- Added Class Service.
- Added various data types for use with Class Service.
- Updated Appointment data structure. Removed "Open" field.

April 3, 2008

- Updated documentation to version 0.4
- Changed all methods to be up to date with version 0.4

April 1, 2008

- Added createLoginAndPassword_HTTP and createLoginAndPassword_SOAP.
- Added updatePassword_HTTP and updatePassword_SOAP.

March 27, 2008

- Added getCardIDHistory_HTTP and getCardIDHistory_SOAP.
- Added Result_StringList.

March 21, 2008

- Added documentation on ShowResources in Class Schedule API.

March 13 2008

- Updated Data Structures: Client, Appointment.
- Added ClientEvent in Outbound Schemas.

March 12 2008

- Renamed getAppointmentByClientID to getAppointmentByCardID
- Changed input parameter int ClientID on getAppointmentByClientID to string CardID.

March 7 2008

- Added API Use with Multiple Studios section in API End User Guide.
- Added [Studio ID Wild Card Compatible] to getAppointmentByClientID methods.
- Added Documentation Change Log section in API End User Guide.

MINDBODY Web Service API Overview

The MINDBODY Web Service API is a system for allowing 3rd parties to programmatically access MINDBODY business logic and data. Security and access are ensured through access control measures and accountability is tracked via an auditing system.

The API's primary goal is to allow easy bi-directional integration with other software providers as well as allowing integration with studio web sites.

Our API is developed using industry standard technologies (XML web services, SOAP, WSDL) to allow the broadest possible compatibility with the least amount of development time for both MINDBODY developers and 3rd party developers.

In summary, when another business entity needs the ability to programmatically read or write data from MINDBODY we offer the MINDBODY Web Service API as our solution.

Quick Start

Request source credentials

The MINDBODY Web Service API requires a sourcename / password / studio ID login collectively referred to as 'source credentials' to process requests. In most cases, credentials will only need to be issued for access to a single MINDBODY site. In this case, the owner of the studio can issue source credentials, and then use it themselves, or give it to a third party developer to allow them access to their data.

The process to issue these credentials are the following:

1. The studio owner logs into their MINDBODY online site as 'owner'
2. Go to Toolbox->Setup->Staff->Users & Groups.
3. Click on 'Issue API Credentials'.
4. The studio owner inputs a password that will be passed to the API (there are complexity requirements on this password).
5. The studio owner clicks on 'Issue Credentials'.

Once the credentials are issued, the sourcename, password, and studioID will be accepted by the API.

If multiple MINDBODY online sites (regions) are going to be accessed through the API, then you must request the source credentials from MINDBODY. To request sourcename credentials, contact MINDBODY at api@mindbodyonline.com.

Please be sure to include:

- Business name
- Developer contact information
- Studio owner contact information
- Desired use of the API
- Studios to be accessed

This information is necessary to confirm the studio owner is allowing access to their data. When your request is processed you will be given a sourcename, password, studio ID list and permissions list. This will let you know what methods you can access as well as what studio's you can run the methods against.

Request method permissions

The MINDBODY Web Service API permission system only allows methods to be accessed by a sourcename if that sourcename has been given explicit permission to run that method. Permissions restrict / allow a sourcename to a method as well as restrict / allow a sourcename to execute a method on a particular studio

(determined by the studio ID). Permission to all methods is granted by default, but can be restricted as requested.

To request a permissions change, contact MINDBODY at api@mindbodyonline.com.

Please be sure to include:

- Business name
- Developer contact information
- Studio owner's contact information
- Desired permissions
- Studios to be accessed

When your request is processed you will receive a confirmation of your updated permissions.

Develop Application and Test

For the location of the current web services, the developer should visit <http://www.mindbodyonline.com/api>. The provided links will list available web services and provide their WSDL document to code against.

When viewed in a browser, each web service provides a list of available methods, and the most recent documentation on their function. Clicking on a method will bring up a detailed view of that method's function, along with expected input parameters and output results. Some calls allow posting data through this page for simple testing. MINDBODY also suggests the use of third party web service testing programs for initial testing.

Currently there is no sandbox application available to test code outside of a live environment. However, using the MINDBODY application, it is very simple to test web service applications and undo any undesired changes.

Some clients have found it useful to create a client specifically for testing purposes in the MINDBODY application. They then use the web service with this test client as the target of all operations. The studio owner can also create a restricted staff login for the developer, to allow the developer to log into the studio's MINDBODY site and delete any sales, etc attached to the test client.

Systems Architecture

Runtime Platform

The platform the MINDBODY Web Service API runs on is the ASP.Net platform. MINDBODY uses ASP.Net web services as not only are they standards compliant but the consumption of ASP.Net web services is heavily documented in books and websites.

Service Availability

The MINDBODY Web Service API resides on the existing high availability web farm. Thus, the uptime expectancy for the MINDBODY Web Service API is identical to the main MINDBODY web application.

preview.mindbodyonline.com and clients.mindbodyonline.com

MINDBODY publishes code to 2 sub-domains. preview.mindbodyonline.com is essentially ‘beta’ code while clients.mindbodyonline.com is considered released production code.

Web Service code is also published through this system so that going to:

http://preview.mindbodyonline.com/api/0_4/ClientService.asmx

may end up running different code than:

http://clients.mindbodyonline.com/api/0_4/ClientService.asmx.

Remember, the newest ‘beta’ code gets published to the preview sub domain and the final released code gets published to the clients sub domain.

It is recommended you consume the “clients” sub domain web services.

Versioning

End-users will write software to consume MINDBODY Web Services. If MINDBODY makes changes to an existing web service that changes any method definitions it may break code written by end-users. This can provide an extremely costly and frustrating experience for the end-user.

To prevent breaking 3rd party code, each new release of the API will not over-write existing API methods but will instead create a separate version instance. Simply, new code is deployed into a new version folder. End-users can either migrate their code to the new version or continue running the older version.

An example of this versioning method is displayed below:

http://clients.mindbodyonline.com/API/0_1/Client.asmx
http://clients.mindbodyonline.com/API/0_2/Client.asmx
http://clients.mindbodyonline.com/API/0_3/ClientService.asmx
http://clients.mindbodyonline.com/API/0_4/ClientService.asmx

WSDL

The WSDL document for the MINDBODY Web Services can be obtained by adding a “? WSDL” to the address of the web service.

An example of retrieving the WSDL is displayed below:

https://clients.mindbodyonline.com/api/0_4/SaleService.asmx?WSDL
https://clients.mindbodyonline.com/api/0_4/ClientService.asmx?WSDL

API Website

An API website has been setup to contain all information needed by 3rd party developers for coding against the MINDBODY Web Service API.

The API website contains the latest up to date information regarding API version updates and any upcoming information of use to 3rd party developers.

<http://www.mindbodyonline.com/api>

Development Assistance

If a 3rd party developer needs help consuming the web service (beyond knowing what the input and output parameters mean) please consult any book or website that deal with consuming SOAP Web Services.

For assistance directly related to the MINDBODY Web Service API, please contact MINDBODY at api@mindbodyonline.com.

Example MBO WS API Consumer Application

The following example creates a console application in .Net 2.0 that consumes the getClientHasMembership MINDBODY Web Service API method.

1. Create a new console application in Visual Studio named WebService_Demo.
2. Right click on the project and select “Add Web Reference”.
3. For the web reference enter the URL http://preview.mindbodyonline.com/api/0_4/ClientService.asmx. When the reference is found in the Visual Studio browser, select to add.
4. Paste the following code into your Program.cs file.

```
using System;
using System.Collections.Generic;
using System.Text;
using WebService_Demo.com.mindbodyonline.preview; //Note the addition of this using statement.

namespace WebService_Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            Demo _Demo = new Demo("Test", "password", -338);
            Console.WriteLine(_Demo.IsMember("1234600"));
            Console.ReadLine();
        }
    }

    class Demo
    {
        ClientsService _ClientService = new ClientsService();
        CredentialsHeader _Credentials = new CredentialsHeader();

        public Demo(string Sourcename, string Password, int StudioID)
        {
            //Setup the credentials header object with our login criteria.
            _Credentials.SourceName = Sourcename;
            _Credentials.Password = Password;
            _Credentials.StudioID = StudioID;

            //Assign our credentials to the web service.
            //When we make SOAP calls the credentials will automatically be provided.
            _ClientService.CredentialsHeaderValue = _Credentials;
        }

        public string IsMember(string RSSID)
        {
            string _ReturnValue = string.Empty;

            try
            {
                //Make the actual web service call.
                Result_Bool mobjResult = _ClientService.getClientHasMembership_SOAP(RSSID);
            }
        }
    }
}
```

```
//Check if the request generated an error such as invalid credentials.
if (mobjResult.Error != ErrorCode.None)
{
    _ReturnValue = "Error while checking membership. Error: " +
mobjResult.Message;
}
else
{
    //Check if we got a positive response...
    if (mobjResult.Result1)
    {
        _ReturnValue = "Valid membership!";
    }
    else
    {
        _ReturnValue = "Invalid membership!";
    }
}
}
catch (Exception e)
{
    //Catch any application errors such as making a request when we
    //there is no network connectivity, etc.
    _ReturnValue = "Error while checking membership. Error: " + e.Message;
}
return _ReturnValue;
}
}
```

When you run the console application you should see a message indicating that the supplied string linked to a client with a valid membership (assuming of course the sourcename, password, studio ID and passed RSSID are all valid).

Method List

The following is a complete listing of methods for API version 0.4.

Appointment Service

http://clients.mindbodyonline.com/api/0_4/appointmentservice.asmx

The Appointment Services provide methods for querying appointment records.

getAppointment

Description: Retrieves an appointment. Appointment record is returned in a Result_Appointment data type which also contains error codes and/or messages in relation to retrieving the appointment record.

If no appointment can be found, an empty appointment record is returned in a Result_Appointment data type. The empty appointment record will have an appointment ID of 0.

Input: int Appointment ID, string Source Name, string Source Password, int Studio ID.

Output: Result_Appointment containing the appointment record on success or an empty appointment record with an appointment ID of 0 and an error message on failure.

getAppointmentByCardID

[Studio ID Wild Card Comptaible]

Notes: Renamed from getAppointmentByClientID_HTTP

Description: Retrieves all appointment records for the specified client. Retrieves records for past and future appointments based on submitted criteria (DaysOfApptsFuture / DaysOfApptsPast) in relation to current date. Appointment records are returned in a Result_Appointments data type which also contains error codes and/or messages in relation to retrieving the appointment records.

Input: string Card ID, int Days Of Appointments In The Future, int Days Of Appointments In The Past, string Source Name, string Source Password, int Studio ID.

Output: Result_Appointments containing the appointment records on success or an empty appointment record with an appointment ID of 0 and an error message on failure.

Class Service

http://clients.mindbodyonline.com/api/0_4/classservice.aspx

getClassesWithClientInfoByCardID

[Studio ID Wild Card Comptaible]

Description: Retrieves all class records for the specified client. Retrieves records for past and future classes based on submitted criteria (DaysOfClassesFuture / DaysOfClassesPast) in relation to current date. Class records are returned in a Result_ClassesWithClientInfo data type which also contains error codes and/or messages in relation to retrieving the class records. Studio ID wild card compatible.

Input: string Card ID, int Days Of Classes In The Future, int Days Of Classes In The Past, string Source Name, string Source Password, int Studio ID.

Output: Result_ClassesWithClientInfo containing the class records with payment information on success or an empty result and an error message on failure.

getClasses

[Studio ID Wild Card Comptaible]

Description: Retrieves a set of classes based on date range. Class record is returned in a Result_Classes data type which also contains error codes and/or messages in relation to retrieving the class record.

If no class can be found, an empty result set is returned in a Result_Classes data type.

Input: datetime DateStart, datetime DateEnd, string Source Name, string Source Password, int Studio ID.

Output: Result_Classes containing the classes' records on success or an empty result set and an error message on failure.

getClass

Description: Retrieves a specified class by the class ID

Input int Class ID, datetime Class Date, string Source Name, string Source Password, int Studio ID.

Output: Result_Class containing the class record on success or an empty result set and an error message on failure.

getClassWithClientInfo

Description: Gets a specific class with payment information for the specified client.

Input int Class ID, int ClientID, datetime Class Date, string Source Name, string Source Password, int Studio ID.

Output: Result_ClassesWithClientInfo containing the class record with payment data on success or an empty result set and an error message on failure.

Client Service

http://clients.mindbodyonline.com/api/0_4/clientservice.asmx

The Clients Service provides methods for adding, modifying or querying client records.

addArrivalEntry

Description: Adds an arrival record for the specified client at the specified location. Arrival records are viewable in the Entry Logs report.

The location field indicates which location the arrival is to be recorded at if the studio has multiple locations using the same database. For single location studios the location can be set to 1 to indicate the primary location.

If the studio has multiple locations sharing the same database, contact MindBody to request a list of LocationIDs for the studio. Be sure to specify the request is in regards to the API.

Input: string Card_ID, int LocationID, string Sourcename, string Password, int StudioID

Output: Result_Bool indicating if the query was run successfully.or detailed error information.

Errors:

InvalidCredentials

InvalidParameters

“Card_ID does not map to a valid ClientID”

addClient

Description: Adds a client to the database and returns the client ID on success. Country and state can be left blank - they will be replaced by the defaults as set in MINDBODY. Card_ID maps to the RSSID field in the MINDBODY database. If Constant Contact integration is enabled, will add to the appropriate Constant Contact list.

Input: string Card_ID, string FirstName, string MiddleName, string LastName, string AddressLine1, string AddressLine2, string City, string State, string PostalCode, string Country, string HomePhone, string CellPhone, string Email, string Birthday, bool Male, bool EmailOptIn, string Sourcename, string Password, int StudioID

Output: Result_Int struct containing the results of the query (the client ID of the added client on success) or detailed error information.

Errors:

InvalidCredentials

InvalidParameters

“RSSID submitted is already in use. You must provide a unique RSSID to assign to the client being added.”

InternalException

“Max ClientID could not be retrieved.”

“Client record could not be inserted.”

getClient

Description: Accepts a Card_ID (i.e. RSSID) and returns a Client_Result containing a client record on success and an error message on failure.

Input: string Card ID, string Source Name, string Source Password, int Studio ID.

Output: Result_Client struct containing the results of the query or detailed error information.

Errors:

InvalidCredentials

InternalException

“Error retrieving ClientID from passed in Card_ID(RSSID)”

InvalidParameters

“Card_ID does not map to a valid ClientID”

getClientFitRewards

Description: For fitRewards! only. Accepts a Card_ID (i.e. MembershipID) and returns a Client_Result containing a client record on success and an error message on failure. Card_ID is in the format of Studioidentifier_MembershipID (example: SampleYoga_01234567).

Input: string Card ID, string Source Name, string Source Password, int Studio ID.

Output: Result_Client struct containing the results of the query or detailed error information.

Errors:

InvalidParameters

“The supplied Card_ID was not prefixed with a studio identifier. Example:

MyYoga_00012301”

“The supplied Card_ID contained a studio identifier that is not linked to any studio.”

getClientHasContract

Description: Accepts a CardID (also known as an RSSID) and checks if the associated client has a valid, current, non-suspended, non-deleted contract. Returns a boolean result on success or a false and error message on failure.

For information on contracts, consult the MINDBODY documentation here:

http://www.mindbodyonline.com/HelpFiles/MyWebHelp/Content/Troubleshooting/Series_Contract_AutoPay_Package.htm

Input: string Card ID, string Source Name, string Source Password, int Studio ID.

Output: Result_Bool struct containing the results of the query or detailed error information.

Errors:

InvalidCredentials

InternalException

“Error retrieving ClientID from passed in Card_ID(RSSID)”

InvalidParameters

“Card_ID does not map to a valid ClientID”

getClientHasMembership

Description: Accepts a CardID (also known as an RSSID) and checks if the associated client has a valid and current membership. Returns a boolean result on success or a false and error message on failure.

For information on memberships, consult the MINDBODY documentation here:
http://www.mindbodyonline.com/HelpFiles/MyWebHelp/Content/Troubleshooting/Series_Contract_AutoPay_Package.htm

Input: string Card ID, string Source Name, string Source Password, int Studio ID.

Output: Result_Bool struct containing the results of the query or detailed error information.

Errors:

InvalidCredentials

InternalException

“Error retrieving ClientID from passed in Card_ID(RSSID)”

InvalidParameters

“Card_ID does not map to a valid ClientID”

getClientTypeList

Description: Retrieves the list of client types (also known as student types) that can be assigned to a client.

Input: string Source Name, string Source Password, int Studio ID.

Output: Result_ClientTypeList struct containing the results of the query or detailed error information.

Errors:

InvalidCredentials

getCardIDHistory

Description: Retrieve the current Card_ID along with the history of Card_IDs based on the supplied Card_ID. Current Card_ID is first element. Original Card_ID is the last element.

Input: string Card ID, string Source Name, string Source Password, int Studio ID.

Output: Result_StringList containing the results of the query or detailed error information.

Errors:

InvalidCredentials

InvalidParameters

“No history could be located for the Card_ID: (Card_ID)”

getClientIndexes

Description: Returns a Result_ClientIndexList containing a list of active client indexes and their associated child client index values. To assign a client index to a client use the desired client index value and Card_ID with updateClientIndexOnClient.

Input: string Source Name, string Source Password, int Studio ID.

Output: Result_ClientIndexList array containing the list of active client indexes and their associated child client index values. In the case of an error, a detailed error message is returned.

Errors:

InvalidCredentials

InternalException

InvalidParameters

“Invalid ClientIndexValueID.”

getClientIndexesOnClient

Description: Returns a Result_ClientIndexList containing a list of all active client indexes and client index values assigned to the specified client.

Input: string Card_ID, string Source Name, string Source Password, int Studio ID.

Output: Result_ClientIndexList array containing the list of active client indexes and their associated child client index values assigned to the specified client. In the case of an error, a detailed error message is returned.

Errors:

InvalidCredentials

InternalException

InvalidParameters

“Card_ID does not map to a valid ClientID”

updateClient

Description: Updates a client record with the given profile information. Unspecified fields will be set to null. The 'ClientNotFoundAction' parameter specifies the action to be taken if a match for the specified Card_ID is not found: a value of 'Add_New' will insert a new client record, a value of 'Fail' will return a failure message. Boolean fields must have a value. State and Country fields must correspond to postal abbreviations (ex: CA, US).

Input: string Sourcename, string Password, int Studio ID, ClientNotFoundAction (Fail | Add_New), string Card_ID, string FirstName, string MiddleName, string LastName, string AddressLine1, string AddressLine2, string City, string State, string PostalCode, string Country, string HomePhone, string CellPhone, string Email, string Birthday, bool Male, bool EmailOptIn.

Output: Result_Boolean indicating if the query ran successfully. In the case of an error, a detailed error message is returned.

Errors:

InvalidCredentials

InternalException

InvalidParameters

“Card_ID does not map to a valid ClientID”

updateClientIndexOnClient

Description: Assigns or removes a client index value to the specified client. Returns a Result_Boolean indicating if the client was updated successfully. Use getClientIndexes to get a list of valid client index values.

Input: string Card_ID, int ClientIndexValueID, bool Assign, string Source Name, string Source Password, int Studio ID.

Output: Result_Boolean indicating if the query ran successfully. In the case of an error, a detailed error message is returned.

Errors:

InvalidCredentials

InternalException

InvalidParameters

“Card_ID does not map to a valid ClientID”

“Invalid ClientIndexValueID.”

updateClientTypeOnClient

Description: Assign (or remove) a client type to a client record. Retrieve valid client types by using getClientTypeList. Returns a Result_Boolean indicating if the query ran successfully.

Input: string Card_ID, int TypeID, bool Assign, string Source Name, string Source Password, int Studio ID.

Output: Result_Boolean indicating if the query ran successfully. In the case of an error, a detailed error message is returned.

Errors:

InvalidCredentials

InternalException

InvalidParameters

“Card_ID does not map to a valid ClientID”

updateLoginNameAndPassword

Description: Updates a client record with the specified loginname and password to be used for logging into MindBodyOnline. Will not update if loginname is already in use by this or any other client and / or password complexity requirement not met.

Input: string Card ID, string LoginName, string LoginPassword, string Source, string Password, int Studio ID.

Output: Result_Boolean indicating if a client record was updated with the new login and password.

Errors:

InvalidCredentials

InvalidParameters

“Card_ID does not map to a valid ClientID”

“Password does not meet complexity requirements. Make sure the password is six or more characters and contains both letters and numbrs.”

“Specified login already exists.”

updatePassword

Description: Updates a client record with the specified password. Will not update if password does not meet complexity requirement.

Input: string Card ID, string LoginPassword, string Source, string Password, int Studio ID.

Output: Result_Bool indicating if a client record was updated with the new password.

Errors:

InvalidCredentials

InvalidParameters

“Card_ID does not map to a valid ClientID”

“Password does not meet complexity requirements. Make sure the password is six or more characters and contains both letters and numbers.”

ValidateUserLogin

Description: Validates a consumer mode login and password for an end user. It also creates a GUID that can be used to log into a MINDBODY site automatically.

Input: string Sourcename, string Password, int Studio ID, string UserName, string UserPassword.

Output: ValidateUserLoginResult containing the matching client record that was validated, and a GUID that can be used to create a link to the online site that will automatically log that user in. An example is shown below:

Link to a site with studioID = XXXX and GUID = YYYY:
clients.mindbodyonline.com/ws.asp?studioid=XXXX&guid=YYYY

The GUID is valid for 30 mins, and the system will automatically log the user in within that time. An invalid GUID will load the online site as normal, but without a logged in user. Other querystring variables in the link can be used, and are generated in the business mode of the online software under toolbox->setup->links & shortcuts.

Errors:

InvalidCredentials

InvalidParameters

“Username and password combination not found”

Region Service

http://clients.mindbodyonline.com/api/0_4/saleservice.asmx

The Region Service provide methods for querying information specific to the region (the online site).

getLocations

Description: Gets a list of active locations for the input StudioID.

Input: string Sourcename, string Source Password, int StudioID, enum ScheduleType

Output: Result_Locations object containing a list of locations in the studio

Method Specific Errors: None

Sales Service

http://clients.mindbodyonline.com/api/0_4/saleservice.asmx

The Sales Service provides methods for adding, modifying or querying sales records. It also provides methods to manipulate a shopping cart object, and pass in credit card information to purchase all items in the cart.

addSaleSimple

Description: Makes a simple inventory sale of retail items or services only. Returns a SaleID on success or a -1 and error message on failure. EmployeeID maps to a TrainerID in the MINDBODY database schema. SaleDateTime must include both a date and time. LocationID should be 1 for the default location, 98 for the online store or 0 for all locations (used in the Sales report to display location of sale). Recorded total will be ((Product Price * Product Quantity) * (1 + Tax rate)). Product Price and Tax Rate are retrieved from the database using the passed BarcodeID.

Input: string Card ID, int Employee ID, datetime Sale Date Time, paymentmethod Payment Method, double Payment Amount, int Location ID, string Barcode ID, int Product Quantity, string Source Name, string Source Password, int Studio ID

Output: Result_Int struct containing the SaleID on success or a -1 and an error message on failure.

Method Specific Errors:

InvalidParameters

“PaymentAmount (Amount) does not equal calculated total (Total).”

addCourseAndPaymentToCart

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

addEnrollmentAndPaymentToCart

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

addItemToCart

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

checkoutShoppingCart

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

getServicePrograms

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

getServices

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

getServicesAndDepositsForCourse

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

getServicesAndDepositsForEnrollment

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

getShoppingCart

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

removeItemFromCart

Description:

Input: string Source Name, string Source Password, int StudioID

Output: Result_ object containing

Method Specific Errors:

Select Services

http://clients.mindbodyonline.com/api/0_4/SelectService.aspx

SelectData

Description: Retrieves requested data based on an input SQL Select query. Because MINDBODY does not document its database schema, consultation with a MINDBODY API developer will be necessary to generate a SQL script that queries the desired data. Select statements are the only scripts allowed to run, and not all tables have permitted access. Scripts that write data can not be run (Update, Delete, etc...).

Input: string SelectSQL, string Source Name, string Source Password, int Studio ID.

Output: Result_RecordSet containing the requested data.

[NOTE] The XML schema of the output is dynamic, since it depends on the values in the SELECT clause of the input SQL string; therefore, the definition of the response object can not be meaningfully parsed by common web service consuming programs (like WSDL.exe). The XML response will have to be traversed manually for this web method.

Staff Service

http://clients.mindbodyonline.com/api/0_4/StaffService.asmx

getStaffList

Description: Retrieves requested data based on an input SQL Select query. Because MINDBODY does not document its database schema, consultation with a MINDBODY API developer will be necessary to generate a SQL script that queries the desired data. Select statements are the only scripts allowed to run, and not all tables have permitted access. Scripts that write data can not be run (Update, Delete, etc...).

Input string Sourcename, string Password, int Studio ID, Tertiary IsClassInstructor, Tertiary IsAppointmentInstructor, Tertiary IsSalesRep, Tertiary IsMale.

Output: Result_StaffList containing the requested data. This contains a list of Trainer records.

Outbound Schemas

http://clients.mindbodyonline.com/api/0_4/outbound.asmx

The Outbound Schemas are not for consumption. They are meant to provide the schema that MindBody is expecting when the MINDBODY software makes an outbound web service call to notify other entities of events in the MINDBODY software.

The purpose is to allow MINDBODY to automatically notify your web service of an event in the MINDBODY web service. Using the parameters passed into your implementation of the web service; make a call back to the MINDBODY web service to get the details of the event.

An example would be notification of appointment events. The goal is to have MINDBODY notify your system of an appointment event, then using the details provided in the notification make a call back to the MINDBODY API to retrieve details about the appointment. The first step would be to implement the AppointmentEvent web service so that upon receiving a call from MINDBODY a separate request would be made from your system to the getAppointment web service method to retrieve information about the appointment. The second step would be to contact MINDBODY at api@mindbodyonline.com of your webservice so that the appropriate code can be added to the MINDBODY system to call your webservice when applicable events occur.

If you want MINDBODY to automatically notify you of events, contact MINDBODY at api@mindbodyonline.com.

AppointmentEvent

Description: Create a webservice with this schema and notify MINDBODY if you want your system to be notified of appointment events. Use the parameters passed to your webservice to call the getAppointment methods to retrieve the appointment information.

Input: int Appointment ID, int Studio ID, AppointmentEventInfo Event Info.

Output: None.

ClientEvent

Description: Create a webservice with this schema and notify MINDBODY if you want your system to be notified of client events. Use the parameters passed to your webservice to call the getClientInfo methods to retrieve the client information.

Input: string Card ID, int Studio ID, ClientEventInfo Event Info.

Output: None.

Data Structures

CredentialsHeader

Description: Used to provide credentials to the web service. Create a credentials header and assign to the web service object's CredentialsHeaderField. When a call is made via the web service object's SOAP methods the credentials will automatically be supplied to the server.

Type: class

Fields:

- Sourcename, string
- Password, string
- StudioID, int

Result / Result_Bool / Result_Int / Result_Client / Result_Appointment / Result_Appointments / Result_StringList / Result_ClientIndexList

Description: All web service methods return a class derived from Result. This allows the web service to return an object that not only returns the requested object but an applicable string message and error code. The derived classes start with Result_ and then the data type being returned as the Result field.

When evaluating a Result, check to see if Error is equal to ErrorCode.None.

Type: class

Fields:

- Message, string
- Error, ErrorCode
- Result, (Bool for Bool_Result, Int for Int_Result, etc)

AppointmentEventInfo

Description: Details what action occurred with an appointment.

Type: enum

Values:

- Create
- Update
- Delete

ErrorCode

Description: ErrorCode is an enum that describes an error, if any.

Type: enum

Values:

- None
- InvalidCredentials
- InvalidParameters
- InternalException
- Unknown

PaymentMethod

Description: The PaymentMethod enum describes a payment method.

Type: enum

Values:

- Cash = 1
- Check = 2
- Credit_Amex = 3
- Credit_VisaMC = 4
- Credit_Discover = 5
- Credit_ATM = 6
- Comp_Guest = 7
- Other = 9
- Trade = 15
- Account = 16
- Return = 99
- ACH = 100

Appointment

Description: The appointment data type contains all the fields needed to represent an appointment.

Type: class

Fields:

- AppointmentID, int
- CardID, string

- TrainerID, int
- AppointmentDate, string
- StartTime, string
- EndTime, string
- Resources, string array
- Notes, string
- LastModifiedByEmployeeID, int
- LastModifiedByEmployeeName, string
- LastModifiedTimestamp, string
- StudioID, int
- Status, AppointmentStatusDetail

Class

Description: The class data type contains all the fields needed to represent a class.

Type: class

Fields:

- ClassID, int
- ClassDate, string
- StartTime, string
- EndTime, string
- StartTime, string
- EndTime, string
- ClassName, string
- Description, string
- TypeGroup, typegroup
- ClassLevel, classlevel
- VisitType, visittype
- Resources, resource array
- Studio, studio
- Location, location
- Trainer, trainer
- Subtrainer, trainer

AppointmentStatusDetails

Description: Details the status of an appointment.

Type: enum

Values:

- Booked

- Complete
- EarlyCancel
- LateCancel
- Confirmed
- Arrived
- Missed
- Pending

Client

Description: The Client data type contains all the fields needed to represent a client.

Type: class

Fields:

- RSSID, string
- ClientID, int
- FirstName, string
- MiddleName, string
- LastName, string
- AddressLine1, string
- AddressLine2, string
- City, string
- State, string
- PostalCode, string
- Country, string
- HomePhone, string
- CellPhone, string
- WorkPhone, string
- WorkExtension, string
- Pager, string
- FaxNumber, string
- Email, string
- Birthday, string
- ReferredBy, string
- SignupDate, string
- FirstClass, string
- Location, int
- HomeStudio, int
- ClientID, int
- Male, bool

- Inactive, bool
- EmailOptIn, bool

ClientType

Description: The ClientType struct represents a possible ClientType—the human readable TypeName and the TypeID.

Type: struct

Fields:

- TypeName, string
- TypeID, int

Product

Description: The product class represents a line item on an invoice. When sent to a web service the web service retrieves information about the product based on the BarcodeID and determines total, subtotal and tax based on the product's price, tax and quantity.

Type: class

Fields:

- Quantity, int
- BarcodeID, string

VisitType

Description: A class visit type.

Type: class

Fields:

- VisitTypeName, string
- VisitTypeID, int

TypeGroup

Description: A typegroup.

Type: class

Fields:

- VisitTypeName, string
- VisitTypeID, int

Trainer

Description: A trainer.

Type: class

Fields:

- TrainerName, string
- TrainerID, int
- ExternalID, string
- FirstName, string
- LastName, string
- DisplayName, string
- Bio, string
- ImageURL, string
- IsDeskStaff, bool
- IsSalesRep, bool
- IsClassInstructor, bool
- IsAppointmentInstructor, bool
- IsMale, bool

Studio

Description: A studio.

Type: class

Fields:

- StudioName, string
- StudioID, int

SeriesInfo

Description: Information about a payment for a class.

Type: class

Fields:

- ProductID, int
- SeriesName, string
- SeriesID, int
- SeriesType, SeriesType
- ExpirationDate, string
- Remaining, int
- RealRemaining, int

SeriesType

Description: The type of payment for a class.

Type: enum

Values:

- ClientCredit = 0
- Count = 1
- Time = 2
- Membership = 3
- Unpaid = 9

DateTimeFormat

Description: The date time format.

Type: enum

Values:

- ShortDate
- LongDate
- ShortTime
- LongTime

ClassVisitStatusDetail

Description: The type of payment for a class.

Type: enum

Values:

- ClientCredit = 0
- Count = 1
- Time = 2
- Membership = 3
- Unpaid = 9

Location

Description: A location.

Type: struct

Values:

- LocationName, string
- LocationID, int

ClientType

Description: A location.

Type: struct

Values:

- TypeName, string
- TypeID, int

ClientIndex

Description: ClientIndexes allow the studio to assign values to a client for categorical purposes. ClientIndexName is the name of the category and ClientIndexValue is a list of the values in the category.

Type: class

Values:

- ClientIndexName, string
- ClientIndexID, int
- ClientIndexValue, array of ClientIndexValue

ClientIndexValue

Description: ClientIndexes allow the studio to assign values to a client for categorical purposes. ClientIndexName is the name of the category and ClientIndexValue is a list of the values in the category.

Type: class

Values:

- ClientIndexValueName, string
- ClientIndexValueID, int

API Use with Multiple Studios

All API methods allow the specification of a studio ID to run the query against. This allows a developer with multiple studios to call a method for an individual studio or to repeatedly call a method for each studio ID they want to query.

Studio ID Wild Card

Some API methods allow the use of a studio ID wild card to specify all studio IDs the source has permissions to query. These methods are noted in the documentation as being “[Studio ID wild card compatible]”. This means that to run a single method against every studio ID the source has permissions on the developer would use the studio ID wild card character in the studio ID field instead of an actual studio ID.

The studio ID wild card character is zero (0).

Queries that are studio ID wild card compatible return their results with each record being marked with the studio ID it originated from. An example would be `getAppointmentByClientID` which can search multiple studios. Each appointment record returned in the result has a studio ID field to show where that record originated from.

Class Schedule API

The Class Schedule API was developed before the MINDBODY Web Service API and uses a different method of access and control. The same functionality is currently available through the MINDBODY Web Service API, and the following is considered deprecated. We suggest using the Web Services for new development, and migrating off of the Class Schedule API to utilize the expanded feature set of the Web Services.

The sourcename / password used for the MINDBODY Web Service API will not work for the Class Schedule API. To request a Class Schedule API sourcename and password, please contact MINDBODY at api@mindbodyonline.com.

Field Definitions

* denotes required:

sourceID* - Assigned by MindBody. Unique to each source, requires matching passcode.

passcode* - This, along with your sourceID assigned to you by MindBody, is a 'login' for the API. There can be more than one valid passcode per sourceID.

startDate* - Date to begin searches

endDate - Date to end searches (defaults to startDate if none provided)

startTime - Time to begin searches (\geq startTime)

endTime - Time to end searches (\leq endTime)

daySun, dayMon, dayTues, dayWeds, dayThurs, dayFri, daySat - bit values (1 = true, 0 = false) for including day of week searches. These are OR'd together, so $\text{daySun}=1 \& \text{dayWeds}=1$ will get you any class that is offered on Sunday OR Wednesday (or Sun/Mon/Thurs or Weds/Fri, etc)

userString - user inputted string (the & character should now work as well) that will match against TypeGroup, ClassDescription, Trainer FirstName or Trainer LastName. **resultType**: Type of results to return. 1 is standard class schedule, 2 is Trainer list, 3 is Location List, 4 is TypeGroup List, 5 is Visit Type List, and 6 is Class Level List.

searchType* - 1, 2, 3 or 4 corresponding to: 1 - Search by Postal Code, 2 - Search by RegionID, 3 - Studio List Search, or 4 - Single Studio Search

Each option below this is based on the searchType.

Search by Postal Code

postalCode* - Postal code to center searches on

range - number of miles to search with postalCode as the center. Valid range of 0 - 50 miles

Search by RegionID

(RegionID search is only useful to select people.)

regionID* - RegionID to search through

Studio List Search

StudioList* - Comma delimited list of studioShorts (e.g. YogaWorksSQLLA, YogaWorksOC, BeYogaSQL). Searches will be run on all webschedulers in the list.

showResources – 0, 1: If set to 1 will return the resource assigned to that class.

Single Studio Search

StudioID* - StudioID of the studio to search. This can be replaced by StudioList from above, and one or the other is required.

typeGroupID - Comma delimited list of TypeGroupIDs to include in the search. (e.g. 1,2,7,10)

visitTypeID - Comma delimited list of visitTypes (Styles) to include in the search. (e.g. 1,2,7,10)

levelID - Comma delimited list of class levelIDs to include in the search. (e.g. 1,2,7,10)

locationID - Comma delimited list of class locationIDs to include in the search. (e.g. 1,2,7,10)

trainerID - Comma delimited list of class trainerIDs to include in the search. (e.g. 1,2,7,10)

showResources – 0, 1: If set to 1 will return the resource assigned to that class.

Class list example

Returns all classes from YogaWorks LA that are scheduled on Wednesday the 6th of March, 2007.

[http://dev.mindbodyonline.com/api/getClassSchedule.asp?
sourceID=999&passcode=test&startDate=3/6/2007&endDate=3/13/2007&searchType=3&StudioList=YogaWorksSQLLA&dayMon=0&dayTues=0&dayWeds=1&dayThurs=0&dayFri=0&daySat=0&daySun=0](http://dev.mindbodyonline.com/api/getClassSchedule.asp?sourceID=999&passcode=test&startDate=3/6/2007&endDate=3/13/2007&searchType=3&StudioList=YogaWorksSQLLA&dayMon=0&dayTues=0&dayWeds=1&dayThurs=0&dayFri=0&daySat=0&daySun=0)

(SourceID and passcode are not actual)

Enrollment list example

Returns all enrollments from YogaWorks LA that are scheduled on Wednesday the 6th of March, 2007.

[http://dev.mindbodyonline.com/api/getEnrollmentSchedule.asp?
sourceID=999&passcode=test&startDate=3/6/2007&endDate=3/13/2007&searchType=3&StudioList=YogaWorksSQLLA&dayMon=0&dayTues=0&dayWeds=1&dayThurs=0&dayFri=0&daySat=0&daySun=0](http://dev.mindbodyonline.com/api/getEnrollmentSchedule.asp?sourceID=999&passcode=test&startDate=3/6/2007&endDate=3/13/2007&searchType=3&StudioList=YogaWorksSQLLA&dayMon=0&dayTues=0&dayWeds=1&dayThurs=0&dayFri=0&daySat=0&daySun=0)

(SourceID and passcode are not actual)

Trainer List Example

If you want a list of trainers from all Yoga Works locations on the 23rd of February between 3pm and 6pm, you can use:

[http://clients.mindbodyonline.com/api/getClassSchedule.asp?
sourceID=999&passcode=test&startDate=2/23/2007&endDate=2/23/2007&startTime=15:00:00&endTime=18:00:00&searchType=3&StudioList=YogaWorksSQLLA,YogaWorksOC,BeYogaSQL&resultType=2](http://clients.mindbodyonline.com/api/getClassSchedule.asp?sourceID=999&passcode=test&startDate=2/23/2007&endDate=2/23/2007&startTime=15:00:00&endTime=18:00:00&searchType=3&StudioList=YogaWorksSQLLA,YogaWorksOC,BeYogaSQL&resultType=2)

(SourceID and passcode are not actual)

Substitute Demo

<class>



```
<studio>
  <studioName>MindBody Yoga Demo</studioName>
  <studioId>-563</studioId>
</studio>
<location>
  <locationName>Uptown Studio</locationName>
  <locationId>1</locationId>
</location>
<trainer>
  <trainerFirstName>Alexis</trainerFirstName>
  <trainerLastName>Ramirez</trainerLastName>
  <trainerId>100000059</trainerId>
</trainer>
<subtrainer>
  <trainerFirstName>Brad</trainerFirstName>
  <trainerLastName>Lohr</trainerLastName>
  <trainerId>100000035</trainerId>
</subtrainer>
<classDate>1/23/2007</classDate>
<classStartTime>11:00:00 AM</classStartTime>
<classEndTime>12:00:00 PM</classEndTime>
<classId>307</classId>
<className>Gerrys Kielty</className>
<classDescription>Level 1</classDescription>
<typeGroup>75 Minute Classes</typeGroup>
</class>
```