

|  |
| --- |
| Mobile Authentication Corporation |
| MAC OTP System API |
|  |

|  |
| --- |
|  |

# MAC OTP System API

## Introduction

This document outlines the MAC System’s APIs. There are two implementations of the APIs, first, as a set of .Net 4.5 DLLs, and second, as a set of JavaScript files. Covered are the API calls between the client’s application and the MAC Web Services.

## Overview

The MAC system is implemented as multiple Web Services, each providing functionally for the ***One-Time Password “OTP”*** and the ***“Send Message”*** features.

In this document, it is assumed that there three major participants; the End Users, the Clients (Content provider including the shopping cart) and MAC (the Service Provider). Not addressed are Payment Processors or other entries that make up the online experience since they do not concern the OTP process.

Note: A MAC Client is the owner of the web site the end user is accessing. The Client has registered with MAC and has been issued a Client ID. The Client ID is used in every exchanged with the MAC OTP System.

## One-Time Password “OTP” APIs

There are two types of calls, Calls for Client Managed End Users and Calls for Registered End Users:

* Client Managed End Users are uses where the client (Merchant/Financial Intrusion, originations, etc.) manages the required user information is. Mobile Phone number, Email address, first and last name and passed that information in the web service calls. MAC retains no end user information in its database and does not validate the mobile phone number or email address.
* Registered End Users are users that have registered with the MAC OTP System by either using the self-register application or Client initiated file registration. For more detail refer to the MAC OTP System Registration document.

### .NET 4.5 DLL

The APIs is implemented as a set of calls that format the requests into a REST interface call to the Web services. The REST Interface calls are synchronous where the request waits for the response.

#### Send OTP Methods

* Send OTP To Client Managed End User: The caller must pass the Client ID, the transaction details and the end user’s phone number. This call puts the burden on the client to verify that the end user’s phone is correct. MAC does not maintain any end user information to support this call. This Call returns a unique Request ID generated by the server.
* Send OTP To Registered End User: The caller must pass the Client ID, the end user’s unique identifier “UID” and the end user’s last name. The end user’s UID and last name must be the same as used in the end user registration process. This Call returns a unique Request ID generated by the server.
* Verify OTP: This call is used to verify an OTP. The caller must pass the Request ID as returned by one if the Send OTP calls and the OTP (entered by the end user). This call returns a Valid on Invalid with a reason (no match, time out, too many retries). Invalid responses can be retried. Time out and too many retry responses cannot be retried.

#### Details

/// <summary>

/// Send OTP to Registered End User

/// </summary>

/// <param name="MACURL">URL for MAC services (Service name added by API)</param>

/// <param name="ClientId">String As issued to the Client</param>

/// <param name="UserLastName">String: (as registered)</param>

/// <param name="UID">String: Unique Identifier (as registered)</param>

/// <param name="UserIp">String: (can be null)</param>

/// <param name="TrxType">Used to select OTP label as defined in Client</param>

/// <param name="TrxDetails">Optional: String containing transaction details</param>

/// <returns>String: Request ID / Error</returns>

/// <summary>

/// Send OPT to Client Managed End User (Client provides user's phone and email)

/// </summary>

/// <param name="MACURL">URL for MAC services (Service name added by API) </param>

/// <param name="ClientId">String As issued to the Client</param>

/// <param name="UserEmail">End user's email address</param>

/// <param name="UserPhoneNumber">End user's mobile phone number</param>

/// <param name="UserIp">String (can be null)</param>

/// <param name="TrxType">Used to select OTP label as defined in Client</param>

/// <param name="TrxDetails">Optional String containing transaction details</param>

/// <returns>String: Request ID / Error</returns>

/// <summary>

/// Verify OTP request

/// </summary>

/// <param name="ClientId"></param>

/// <param name="RequestId">Saved from send OTP request</param>

/// <param name="Otp">OTP entered by end user</param>

/// <returns>String: Valid / Invalid (reason) </returns>

#### Send Message Methods

* Send Message to Client Managed End User: The caller must pass the Client ID, the message details and the end user’s phone number.
* Send Message to Registered End User: The caller must pass the Client ID, the end user’s unique identifier “UID”, the end user’s last name and the message text.

#### Details

/// <summary>

/// Send OTP to Registered End User

/// </summary>

/// <param name="MACURL">URL for MAC services (Service name added by API) </param>

/// <param name="ClientId">String As issued to the Client</param>

/// <param name="UserLastName">String: (as registered) </param>

/// <param name="UID">String: Unique Identifier (as registered) </param>

/// <param name="UserIp">String: (can be null) </param>

/// <param name="Message">String containing the text of the message</param>

/// <returns>String: Sent / Error</returns>

/// <summary>

/// Send OPT to Client Managed End User (Client provides user's phone and email)

/// </summary>

/// <param name="MACURL">URL for MAC services (Service name added by API) </param>

/// <param name="ClientId">String As issued to the Client</param>

/// <param name="UserEmail">End user's email address</param>

/// <param name="UserPhoneNumber">End user's mobile phone number</param>

/// <param name="UserIp">String (can be null) </param>

/// <param name="Message">String containing the text of the message</param>

/// <returns>String: Sent / Error</returns>

### JavaScript API

The JavaScript API contains two set of functions, The “Named Fields” functions are tightly linked to the HTML forms and expects the HTML Pages and Field names to be consistent and the functions that expect parameters to be passed in the function calls.

If a client is integrating their HTML pages with the MAC JavaScript API and using the ***Named Fields*** functions they must use the same field named and ***Call Parameters*** functions in their HTML page. MAC does provide client custom HTML pages to assist in the integration process.

#### OTP Named Field functions

These function pull parameters from fields on the html page and send OTP messages to the end user.

function NamedFields\_RequestOtpClientManagedEndUser()

function NamedFields\_RequestOtpRegisteredEndUser()

function NamedFields\_RequestOtpEndUser\_Result(\_response)

function NamedFields\_CancelOtp()

function NamedFields\_ResendOtp()

function NamedFields\_VerifyOtp()

function NamedFields\_VerifyOtp\_Result(\_response)

#### OTP Caller Parameters functions

These functions are called to send OTP messages to the end user.

function RequestOtpClientManagedEndUser(ClientId, EndUserPhoneNumber, EndUserEmail, EndUserIp, LabelIndex, TransactionDetails, CallbackFunction)

Where:

* ClientId – The Unique Id issued by MAC when the Client registered to use the MAC OTP system.
* EndUserPhoneNumber – End User’s 10 digit mobile phone number (9045551212).
* EndUserEmail – End User’s email address.
* EndUserIp – (optional) IP address of End User’s computer, default 0.
* LabelIndex – (optional) Index number for OTP label, see client configuration, default 0.
* TransactionDetails – (optional) Used for transaction verification, see the section on “Text Formatting” for details.
* CallbackFunction – Function to be called to process the response when request completes. Response is in Xml format.

function RequestOtpRegisteredEndUser(ClientId, EndUserUniqueIdentifier, EndUserLastName, EndUser\_IP, LabelIndex, TransactionDetails, CallbackFunction)

Where:

* ClientId – The Unique ID issued by MAC when the Client registered to use the MAC OTP system.
* EndUserUniqueIdentifier – The Unique ID that was used when the End User was registered, see the Unique Id section of this document.
* EndUserLastName – End User’s last name that was user when End User was registered.
* EndUser\_IP – (optional) IP address of End User’s computer, default 0.
* LabelIndex – (optional) Index number for OTP label, see client configuration, default 0.
* TransactionDetails – (optional) Used for transaction verification, see the section on “Text Formatting” for details.
* CallbackFunction – Function to be called to process the response when request completes. Response is in Xml format. Note: contains the RequestId, this must be saved and returned in the VerifyOTP, ResendOtp and CancelOtp calls.

function CancelOtp(ClientId, RequestId, CallbackFunction)

Where:

* ClientId – The Unique Client ID issued by MAC when the Client registered to use the MAC OTP system.
* RequestId – OTP correlation number, returned by the Request OTP calls.
* CallbackFunction – Function to call on response.

function ResendOtp(ClientId, RequestId, CallbackFunction)

Where:

* ClientId – The Unique Client ID issued by MAC when the Client registered to use the MAC OTP system.
* RequestId – OTP correlation number, returned by the Request OTP calls.
* CallbackFunction – Function to call on response.

function VerifyOTP(ClientId, RequestId, Otp, CallbackFunction)

Where:

* ClientId – The Unique Client ID issued by MAC when the Client registered to use the MAC OTP system.
* RequestId – OTP correlation number, returned by the Request OTP calls.
* Otp – The OTP code to be verified.
* CallbackFunction – Function to call on response.

#### Send Message Named Field functions

These function pull parameters from fields on the html page and send text messages to the end user.

function NamedFields\_SendMessageToClientManagedEndUser()

function NamedFields\_SendMessageToRegisteredEndUser()

function NamedFields\_SendMessage\_Result(\_response)

#### Send Text Message Caller Parameters functions

These functions are called to send a text messages to the end user.

function SendMessageToClientManagedEndUser(ClientId, EndUser\_PhoneNumber, EndUser\_Email, EndUser\_IP, Message, CallbackFunction)

* ClientId – The Unique ID issued by MAC when the Client registered to use the MAC OTP system.
* EndUser\_UID – The Unique ID that was used when the End User was registered, see the Unique ID section of this document.
* EndUser\_LastName – End User’s last name that was user when End User was registered.
* EndUser\_IP – (optional) IP address of End User’s computer, default 0.
* Message – Text Message, see the section on “Text Formatting” for details.
* CallbackFunction – Function to be called to process the response when request completes. Response is in Xml format.

function SendMessageRegisteredEndUser(ClientId, EndUserUniqueIdentifier, EndUserLastName, EndUserIp, Message, CallbackFunction)

Where:

* ClientId – The Unique ID issued by MAC when the Client registered to use the MAC OTP system.
* EndUserUniqueIdentifier – The Unique ID that was used when the End User was registered, see the Unique ID section of this document.
* EndUserLastName – End User’s last name that was user when End User was registered.
* EndUuseIp – (optional) IP address of End User’s computer, default 0.
* Message – Text Message, see the section on “Text Formatting” for details.
* CallbackFunction – Function to be called to process the response when request completes. Response is in Xml format.

##### Named Fields, Types, Examples

The following are a list of fields that are expected when using the “Name Field” functions.

1. hiddenClientId – Text, Client’s Id, stored by the client application.
   1. <input type="hidden" id="hiddenClientID" value="????/>

***Note:*** The client id value as issued to the client (merchant/Bank) by MAC when the client registered to use the MAC OTP System.

1. hiddenAction – Text, Event information stored by client application.
   1. <input type="hidden" id="hiddenAction" value="" />

***Note***: This field is used differently based on the function being called. Refer to the MACClientAPI.js file for details.

1. hiddenOtpRequestId – Text, OTP correlation number, stored by client application.
   1. <input type="hidden" id="hiddenOtpRequestId" value="" />

***Note***: This field is used differently based on the function being called. Refer to the MACClientAPI.js file for details.

1. txtUID – Text, The Unique End User’s Id uses when the end user was registered.
   1. <input type='text' id='txtUID' />

***Note***: The end user’s unique identifier that was used in the registration process.

1. txtLastName – Text, The end user’s last name as used in end user registration.
   1. <input type='text' id='txtLastName' />

***Note***: The end user’s last name must be the same as the last name used in the end user registration process.

1. txtMPhoneNo – Text, The end user’s mobile phone number.
   1. <input type='text' id='txtMPhoneNo' />
2. txtEmailAdr – Text, The end user’s preferred email address.
   1. <input type='text' id='txtEmailAdr' />
3. txtIP – Text, The IP address of the end user’s computer.
   1. <input type='text' id='txtIP' />

***Note***: The IP address of the end user’s computer.

1. txtOtpLabelIndex - Text, (optional) Index number for OTP label, refer to client configuration, default 0.
   1. <input type='text' id='txtOtpLabelIndex' />

***Note***: This parameter is used to select the OTP label as configured in the client record.

1. txtOtp – Text, The OTP code to be verified.
   1. <input id='txtOtp' type='text' onclick='javascript: clearInput(this)' value='Enter OTP here' />";
2. txtMessage – textarea, Multi line text input area.
   1. <textarea id='txtMessage' style='border: solid 1px #0000ff; width: 300px; height: 50px; color: #000000;'></textarea>";
3. lbError – <span>, where to write the error messages to be displayed.
   1. <span id='lbError' style='color:Red;'></span>

## Text Formatting

The text of a message must be correctly formatted before it is passed to the API in order to get the desired results.

The Transaction Details must have all “new lines” replaced by pipe ‘|’ characters. The message send routines in the MAC services will replace the pipe characters with the appropriate characters depending on the message deliver method.

Example: A transaction details string of “Item 1 $3.00|Item 2 $1.00|Total $4.00” would be delivered as three lines after the OTP.