

# API Reference

Cueto Event Management System | December 2019

## Table of Contents

Introduction .....	8
Contact Info.....	8
Terms of Use .....	8
Basics.....	9
Obtaining API Credentials .....	9
Connecting to Your Site’s API.....	9
Data Types and Formats .....	10
Date.....	10
Decimal .....	10
GUID (Unique Identifier, UUID).....	10
Int (Integer) .....	10
String .....	10
True/False .....	10
Response Formats.....	10
Generic Action Response .....	11
Specifying Request Action.....	11
HTTP Status Codes .....	12
API Relay Program.....	13
Example Configuration and Calls .....	13
Screenshots.....	14
API Calls: General .....	15
version: Get API Version .....	15
Parameters.....	15
Response.....	15
Sample Call.....	15
Sample Response .....	15
API Calls: Access Control .....	16
acactivity: Get Activity/Scan Records .....	16

Parameters.....	16
Response.....	16
Sample Call.....	17
Sample Response.....	17
aclocations: Get Scanning Locations and Details.....	18
Parameters.....	18
Response.....	18
Sample Call.....	18
Sample Response.....	18
acresults: Get Scan Result Codes and Details.....	19
Parameters.....	19
Response.....	19
Sample Call.....	19
Sample Response.....	19
Current Values.....	20
acscanmodes: Get Scan Modes.....	20
Parameters.....	20
Response.....	20
Sample Call.....	21
Sample Response.....	21
Current Values.....	21
API Calls: Gift Cards.....	22
gcactivitytypes: Get Gift Card Activity Types.....	22
Parameters.....	22
Response.....	22
Sample Call.....	22
Sample Response.....	22
Current Values.....	23
gcadd: Add New Gift Card(s).....	23
Parameters.....	23
Response.....	24
Sample Call.....	24

Sample Response .....	24
gccard: Get Card Details and Activity .....	24
Parameters .....	24
Response .....	24
Sample Call .....	26
Sample Response .....	26
gcdelete: Delete Gift Card .....	27
Parameters .....	27
Response .....	27
Sample Call .....	27
Sample Response .....	27
gcdeleteusage: Delete Gift Card Usage .....	27
Parameters .....	27
Response .....	28
Sample Call .....	28
Sample Response .....	28
gcsearch: Search Gift Cards .....	28
Parameters .....	28
Response .....	28
Sample Call .....	29
Sample Response .....	29
gctemplates: Get Gift Card Templates .....	29
Parameters .....	29
Response .....	29
Sample Call .....	30
Sample Response .....	30
gctypes: Get Gift Card Types .....	31
Parameters .....	31
Response .....	31
Sample Call .....	31
Sample Response .....	31
Current Values .....	31

gcupdate: Update Gift Card Details .....	32
Parameters .....	32
Response .....	32
Sample Call .....	33
Sample Response .....	33
gcuse: Use Card .....	33
Parameters .....	33
Response .....	33
Sample Call .....	34
Sample Response .....	34
API Calls: Ticketing .....	35
otsorder: Get Ticket Order .....	35
Parameters .....	35
Response .....	35
Sample Call .....	39
Sample Response .....	39
otsorders: Get Ticket Orders .....	41
Parameters .....	41
Response .....	41
Sample Call .....	42
Sample Response .....	42
otspackages: Get Ticket Packages .....	42
Parameters .....	42
Response .....	42
Sample Call .....	43
Sample Response .....	43
otstickets: Get Tickets .....	44
Parameters .....	44
Response .....	44
Sample Call .....	45
Sample Response .....	45
API Calls: Volunteers .....	47

volcheckin: Volunteer Shift Check In or Out .....	47
Parameters .....	47
Response .....	47
Sample Call .....	47
Sample Response .....	47
volcommittees: Get Committees and Member Counts .....	47
Parameters .....	47
Response .....	48
Sample Call .....	48
Sample Response .....	48
volpayments: Get Volunteer Payments .....	48
Parameters .....	48
Response .....	49
Sample Call .....	49
Sample Response .....	49
volphotoproblem: Flag/Unflag Bad Photograph .....	50
Parameters .....	50
Response .....	50
Sample Call .....	50
Sample Response .....	50
volprofile: Get Volunteer Profile .....	51
Parameters .....	51
Response .....	51
Sample Call .....	54
Sample Response .....	54
volprofiles: Get Volunteer Profiles .....	56
Parameters .....	57
Response .....	57
Sample Call .....	57
Sample Response .....	57
Code Samples .....	58
C# (.NET 4.7) .....	58

Committee Class Definition .....	58
Sample Code .....	58

## Introduction

This guide is intended to cover the API for the Cueto Event Management Software. The API is under active development based on feedback from customers and revisions will be posted as often as possible.

Is there a new API call or tool that would help you out? Just let us know!

## Contact Info

Cueto & Cueto, Inc. is the company responsible for developing, maintaining and hosting the Cueto Event Management Software. We welcome – and appreciate - your questions and feedback about the software, website and manual.

The best way to get in touch with us is through e-mail at [john@cuetoems.com](mailto:john@cuetoems.com). You can also use our contact form:

<http://www.cuetoems.com/Contact.aspx>

If you're associated with an event, please put the name of the event in the subject of the message so the correct person gets your message. Due to our travel schedule, it may take us several days to respond.

## Terms of Use

This document in its entirety is Copyright 2019 Cueto & Cueto, Inc. It may be copied or distributed, in part or in whole, by any of our customers, or agents acting on their behalf, for purposes of education and training.

This document may not be used for purposes of reverse engineering or duplication of the Cueto Event Management Software.

The most recent version can always be found at:

<http://www.cuetoems.com/Help.aspx>



## Basics

This section covers some basic information, such as obtaining API credentials from Cueto & Cueto, connecting to the API for your copy of the software and data format information for your requests and our responses.

### Obtaining API Credentials

Due to the sensitive nature of information that can be accessed with the API, Cueto & Cueto will only issue API credentials upon request of our primary contact at the event. You can request complete API access or access to one or more of the major modules, such as Access Control, Ticketing and Volunteers, and we'll request information on your expected request rate.

You will be given a username and access key, which must be kept private. Unauthorized distribution of API credentials or an abusive request rate will lead to the termination of API access.

You will include the username and access code as query string variables named *u* and *a*, respectively. You can see an example in the next section.

### Connecting to Your Site's API

You can connect to your event's API using the following address scheme:

<https://www.cuetoems.com/EVENTNAME/api.ashx>

For instance, our demo site which is at:

[https://www.cuetoems.com/cems\\_demo](https://www.cuetoems.com/cems_demo)

Would have the API available at:

[https://www.cuetoems.com/cems\\_demo/api.ashx](https://www.cuetoems.com/cems_demo/api.ashx)

Finally, if you were given the username **api** and access code **385AE3A5-C3AC-4485-9E41-81B15E2E513F**, you would include them in your request like this:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F)

You would then append additional parameters based on the request you're making. You can use an HTTP POST and make sure to use the HTTPS version of the URL. The CEMS server only accepts requests via TLS 1.2.

## Data Types and Formats

Throughout this guide we will reference certain data types when referring to information that you send us or information that we return to you. International conventions, such as using a comma instead of a decimal point, may lead to undesired results so please use US numerical conventions.

### Date

Information representing a date and/or time, with or without a time zone, in any of the various formats that the .NET date and time interpreter can understand:

<https://docs.microsoft.com/en-us/dotnet/standard/base-types/parsing-datetime>

For instance, these are all valid date formats for July 29, 1985:

- 7/29/1985
- July 29, 1985
- 19850729
- 7/29/1985 8:00 PM
- 7/29/1985 8:00 PM GMT-5

### Decimal

A number with or without a decimal point, such as 19.85. Used for all currency values, which may include up to four decimal places.

### GUID (Unique Identifier, UUID)

A globally unique identification number, used for your access code and identifying certain records. An example:

1267B778-F796-424D-98BA-6D32D0AAA80B

### Int (Integer)

A whole number, such as 29, with no formatting or currency symbols. You may or may not include comma separators and any values after a decimal point will be ignored.

### String

Text information, such as the word **document**. Length restrictions will be indicated with the value in parentheses, such as **address (50)**, for a field named **address** with a maximum length of 50 characters.

### True/False

Any of various ways to represent the values true and false, including the strings “true” and “false” and numbers 1 and 0.

## Response Formats

The API can send responses as XML or JSON and you can specify your response format with the **format** query string parameter. To expand the earlier sample URL with your username and access code, you would specify an XML response like this:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml)

Or a JSON response like this:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=json](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=json)

If you don't specify a response format, the API will default to XML.

### Generic Action Response

API calls that perform an action and don't return any data will send back a Generic Action Response object with these fields:

Value	Data Type	Notes
Status	string	OK for a successful call, otherwise Error
Message	string	Filled in if there is an error or if a successful call returns a diagnostic message you might want to log

As XML, a Generic Action Response (successful) will look like:

```
<?xml version="1.0" encoding="utf-8"?>
<Result>
  <Status>OK</Status>
  <Message />
</Result>
```

As JSON, a Generic Action Response (failed) will look like:

```
{ "Status" : "Error", "Message" : "Database error. Please see log." }
```

### Specifying Request Action

You tell the API which function to run using the **action** query string parameter, which we'll give you for each API call in the manual.

For instance, to get a list of all access control locations, you would use the action **aclocations**. Building on the sample URLs, you would request the access control locations in XML like this:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=aclocations](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=aclocations)

Or as JSON like this:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=json&action=aclocations](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=json&action=aclocations)

The order of query string parameters does not matter.

## HTTP Status Codes

The API will return the following status codes:

200 – Successful request

400 – Missing or incorrect action parameter

403 – Missing or incorrect username or access code, insufficient access

500 – Server error; will get logged and reported, but you can let us know if you'd like

## API Relay Program

The CEMS API Relay is a local HTTP server that relays API calls to the CEMS web server. It displays a list of API calls and their results, and includes extensive logging and debugging tools. It can be used:

- To assist with development, by showing you the request as it is seen by the web server and the raw response the API returns.
- To accommodate special situations where an environment may not be able to handle HTTP response codes or TLS 1.2.
- To meet security requirements prohibiting stations from having direct internet access or routing all internet access through a single point.
- To hide your API credentials from client applications (see the configuration section below).

The API Relay requires Windows with the .NET Framework v4.7 or higher installed and is available on request for customers working on an API integration. The computer running the API Relay must support TLS 1.2.

## Example Configuration and Calls

You can specify the target site, port to use, logging level and the API credentials inside the appSettings section of the configuration file. We will provide a default version already set up for your site that looks like this:

```
<appSettings>
  <add key="Port" value="8011" />
  <add key="TargetSite" value="www.cuetoems.com/cems_demo" />
  <add key="Username" value="relay" />
  <add key="AccessCode" value="57724851-0945-4f19-9cd4-f92810fbad8e" />
  <add key="LoggingMode" value="full" />
</appSettings>
```

Since the API credentials are stored in the configuration file, calls sent to the API relay don't need to include them. This can be used to hide your API credentials from client applications if desired.

The API Relay can be referenced via **localhost** or the computer's IP address. For instance, you could request the version number of the API locally on port of 8011 like this:

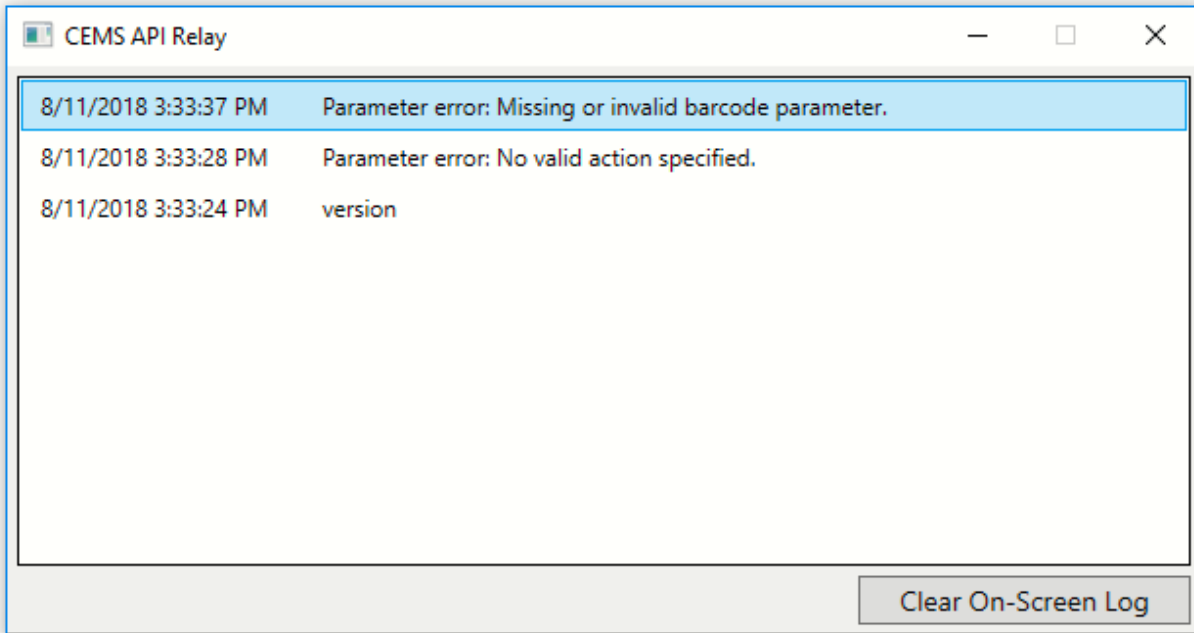
<http://localhost:8011/?action=version&format=xml>

If the computer's IP address is 192.168.1.5 and you have opened up port 8011 on your firewall, other computers could request the version number like this:

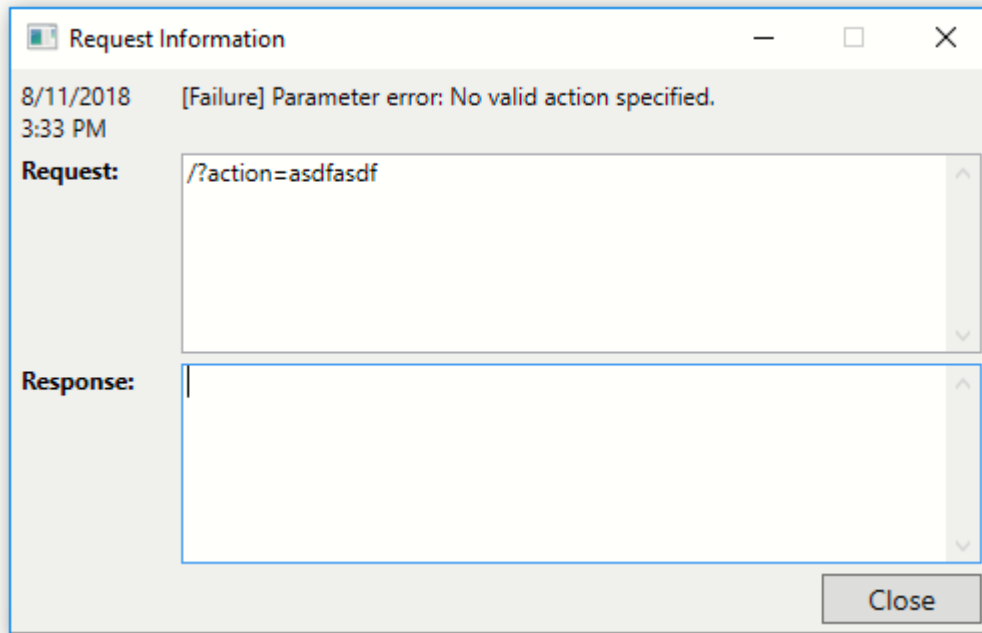
<http://192.168.1.5:8011/?action=version&format=xml>

## Screenshots

While running, the API Relay shows a reverse chronological list of calls, indicating success or failure:



Double clicking on a line will show the overall status, any error messages, and the raw request and response:



## API Calls: General

This section covers all API calls that aren't associated with a specific module.

### version: Get API Version

The API is specific to each copy of the site and the version depends on the software version of the site. Sites for completed events eventually stop being updated. This call gives you the current API version so you know which calls are supported.

#### Parameters

The **action** parameter must be set to **version**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	version	<b>Required</b>

#### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>Version</b>	string(8)	Always returns an 8 character string in the format YYYYMMDD.

#### Sample Call

This API call would request the version as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=version
```

#### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<Version>20191201</Version>
```

## API Calls: Access Control

This section covers all API calls for the access control/gate scanning module.

### acactivity: Get Activity/Scan Records

This call will return a list of ID numbers, date information, barcode information, scan results and order/customer information when available, for each scan in the system that meets your filter criteria.

#### Parameters

The **action** parameter must be set to **acactivity** and you may specify some optional filters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	acactivity	<b>Required</b>
<b>mindate</b>	date	Any date/time format	<b>Optional</b> - Specifies a minimum date/time for records
<b>maxdate</b>	date	Any date/time format	<b>Optional</b> - Specifies a maximum date/time for records
<b>location</b>	int	Any valid location ID	<b>Optional</b> - You can use the <b>aclocations</b> call to get a list of valid IDs
<b>hidedenies</b>	true/false	true/1 or false/0	<b>Optional</b> – If not specified or invalid value, defaults to false

#### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>ActivityID</b>	int	Unique, sequential ID per scan
<b>Date</b>	date	Includes time and time zone
<b>Barcode</b>	string(20)	
<b>LocationID</b>	int	
<b>LocationName</b>	string(50)	
<b>LocationIsEntrance</b>	true/false	Whether the location's scans count towards total attendance
<b>ResultCode</b>	int	
<b>ResultMessage</b>	string	



If the system is able to find ticket order information for the barcode, a TicketInformation section with the following information will be added:

Value	Data Type	Notes
<b>TicketName</b>	string(100)	
<b>OrderNumber</b>	GUID	
<b>FirstName</b>	string(25)	The customer's first name
<b>LastName</b>	string(35)	The customer's last name

If the system is able to match the barcode to a credential, a CredentialInformation section with the following information will be added:

Value	Data Type	Notes
<b>UserID</b>	GUID	The CEMS globally unique user ID
<b>FirstName</b>	string(25)	The person's first name
<b>LastName</b>	string(35)	The person's last name
<b>CredentialName</b>	string(50)	The type of credential

### Sample Call

This API call would request all scan records from July 29, 2016 through July 31, 2016 as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=acactivity&mindate=2016-07-29&maxdate=2016-07-31](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=acactivity&mindate=2016-07-29&maxdate=2016-07-31)

### Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<ScanRecords>
  <ScanRecord>
    <ActivityID>1</ActivityID>
    <Date>2016-07-29T12:23:30.717</Date>
    <Barcode>G00DTIX</Barcode>
    <LocationID>2</LocationID>
    <LocationName>Front Gate</LocationName>
    <LocationIsEntrance>true</LocationIsEntrance>
    <ResultCode>0</ResultCode>
    <ResultMessage>Allowed - Ticket</ResultMessage>
    <TicketInformation>
      <TicketName>Any Day Grounds</TicketName>
      <OrderNumber>61862043-5ba7-4810-80a8-8b7a0d506353</OrderNumber>
      <FirstName>SUDC</FirstName>
      <LastName>Test</LastName>
    </TicketInformation>
  </ScanRecord>
  <ScanRecord>
    <ActivityID>2</ActivityID>
```

```

    <Date>2016-07-30T14:23:32.2</Date>
    <Barcode>BADTIX</Barcode>
    <LocationID>2</LocationID>
    <LocationName>Front Gate</LocationName>
    <LocationIsEntrance>true</LocationIsEntrance>
    <ResultCode>2</ResultCode>
    <ResultMessage>Denied - No Barcode Match</ResultMessage>
  </ScanRecord>
</ScanRecords>

```

## aclocations: Get Scanning Locations and Details

This call will return a list of ID numbers, names and entrance status for each location in the system. This information is also included automatically with the **acactivity** API call.

### Parameters

The **action** parameter must be set to **aclocations**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	aclocations	<b>Required</b>

### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>LocationID</b>	int	
<b>Name</b>	string(50)	
<b>IsEntrance</b>	true/false	Whether the location's scans count towards total attendance

### Sample Call

This API call would request the list of scanning locations as XML:

```

https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=aclocations

```

### Sample Response

```

<?xml version="1.0" encoding="utf-8"?>
<Locations>
  <Location>
    <LocationID>1</ActivityID>
    <Name>Bus Lot</Name>
    <IsEntrance>true</IsEntrance>
  </Location>
  <Location>
    <LocationID>2</ActivityID>

```

```

    <Name>Front Gate</Name>
    <IsEntrance>true</IsEntrance>
  </Location>
</Locations>

```

## acresults: Get Scan Result Codes and Details

This call will return a list of ID numbers and text explanations of the result codes assigned to each scan. This information is also included automatically with the **acactivity** API call.

### Parameters

The **action** parameter must be set to **acresults**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	acresults	<b>Required</b>

### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>Code</b>	int	0 and 1 are considered success, all else are failures
<b>Message</b>	string	First word is always Allowed or Denied, followed by a dash and a longer explanation

### Sample Call

This API call would request the list of result codes as XML:

```

https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=acresults

```

### Sample Response

```

<?xml version="1.0" encoding="utf-8"?>
<Results>
  <Result>
    <Code>0</Code>
    <Message>Allowed - Ticket</Message>
  </Result>
  <Result>
    <Code>7</Code>
    <Message>Denied - Wrong Day's Ticket</Message>
  </Result>
</Results>

```

## Current Values

This API call will always return these results. If we add more result codes in the future, we will add on to the end and not change any existing values, and the API call and guide will be updated.

Code	Message	Notes
0	Allowed - Ticket	
1	Allowed - Credential	
2	Denied - No Barcode Match	
3	Denied - Deny	A manual deny has been set for this barcode
4	Denied - Alert	A manual alert has been set for this barcode
5	Denied - Repeat Entry	
6	Denied - Non-Entry Ticket	An item in the system but not allowed for access, such as a concessions voucher
7	Denied - Wrong Day's Ticket	
8	Denied - Insufficient Access	The ticket or credential type is not allowed in a certain area
9	Denied - Deactivated	The barcode has been deactivated by the event staff

## acscanmodes: Get Scan Modes

This call will return a list of ID numbers and text explanations of the scanning mode used by tickets. This information is also included automatically with the **otstickets** API call.

### Parameters

The **action** parameter must be set to **acscanmodes**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	acscanmodes	<b>Required</b>

### Response

The response will include the following values for each record:

Value	Data Type	Notes
ID	int	
Name	string	

## Sample Call

This API call would request the list of result codes as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=acscanmodes
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ScanModes>
  <ScanMode>
    <ID>0</ID>
    <Name>Unlimited</Name>
  </ScanMode>
  <ScanMode>
    <ID>1</ID>
    <Name>Single Use</Name>
  </ScanMode>
</ScanModes>
```

## Current Values

This API call will always return these results. If we add more scan modes in the future, we will add on to the end and not change any existing values, and the API call and guide will be updated.

ID	Name	Notes
0	Unlimited	Tickets allows any number of scans on any date
1	Single Use	Ticket allows a single scan on any date
2	Once Per Day	Ticket allows one scan every day
3	No Entry (Ever)	Ticket will be denied on scanning
4	No Entry (Wrong Day)	Forces a wrong day deny on scanning
5	Single Use (On Date)	Ticket allows one scan on the associated date
6	Unlimited (On Date)	Ticket allows any number of scans on the associated date

## API Calls: Gift Cards

This section covers all API calls for the gift and discount cards and charge accounts module.

### gcactivitytypes: Get Gift Card Activity Types

This call will return a list of ID numbers and text explanations of the activity types logged by the gift card system. This information is also included automatically with the **gcard** API call.

#### Parameters

The **action** parameter must be set to **gcactivitytypes**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gcactivitytypes	<b>Required</b>

#### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>ID</b>	int	
<b>Name</b>	string	

#### Sample Call

This API call would request the list of activity types as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcactivitytypes
```

#### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Types>
  <ActivityType>
    <ID>0</ID>
    <Name>Used</Name>
  </ActivityType>
  <ActivityType>
    <ID>1</ID>
    <Name>Changed</Name>
  </ActivityType>
  <ActivityType>
    <ID>2</ID>
    <Name>Created</Name>
  </ActivityType>
</Types>
```

## Current Values

This API call will always return these results. If we add more activity types in the future, we will add on to the end and not change any existing values, and the API call and guide will be updated.

ID	Name	Notes
0	Used	Card/account used by the <b>gcuse</b> call
1	Changed	Card value changed through the web interface
2	Created	Card/account added or imported through the web interface

## gcadd: Add New Gift Card(s)

This call will create new gift cards based on the specified parameters and return the details of all newly created cards. Cards can be created with a specific sequence of barcodes or randomly generated for a certain quantity.

### Parameters

The **action** parameter must be set to **gcadd**. There are several required parameters and a number of optional ones. The **mode** parameter determines how barcodes are generated and what other parameters are required.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gcadd	<b>Required</b>
<b>template</b>	int		<b>Required</b> ; ID number of the <a href="#">template</a>
<b>type</b>	int	0 – 2	<b>Required</b> ; ID number of the <a href="#">card type</a>
<b>value</b>	decimal		Not needed for charge accounts, defaults to 0 if invalid or missing
<b>mode</b>	string(1)	r, s	r is used for random barcodes of a certain quantity, s is used for sequential barcodes
<b>quantity</b>	int		<b>Required for random</b>
<b>prefix</b>	string(50)	A - Z	<b>Required for sequential</b> ; total of prefix and sequence is 50 characters max
<b>start</b>	int		<b>Required for sequential</b>

<b>end</b>	int		<b>Required for sequential;</b> must be greater than start
<b>pad</b>	int	0 - 50	<b>Required for sequential</b>

## Response

The response will contain a list of CardInformation objects that were added to the system, which you can find detailed in the **gccard** section below.

## Sample Call

This API call would create five randomly barcoded gift cards with the template ID 2 and request a response as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcadd&template=2&type=0&mode=r&quantity=5
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<AddResults>
  <CardInformation>See the gccard section below for details.</CardInformation>
  <CardInformation>See the gccard section below for details.</CardInformation>
  <CardInformation>See the gccard section below for details.</CardInformation>
  <CardInformation>See the gccard section below for details.</CardInformation>
  <CardInformation>See the gccard section below for details.</CardInformation>
</AddResults>
```

## gccard: Get Card Details and Activity

This call will return details and the activity history for the card with the specified barcode.

## Parameters

The **action** parameter must be set to **gccard** and you must specify the **barcode** parameter.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gccard	<b>Required</b>
<b>barcode</b>	string(50)	0 – 9, a – z, A – Z	<b>Required</b>

## Response

The response will include the following values:

Value	Data Type	Notes
<b>CardID</b>	GUID	The system's globally unique card ID
<b>Barcode</b>	string(50)	
<b>TypeID</b>	int	ID number of the card type



<b>TypeName</b>	string	Friendly name of the card type
<b>TemplateID</b>	int	ID number of the template
<b>TemplateName</b>	string(50)	Friendly name of the template
<b>SingleUse</b>	true/false	Can the card only be used once?
<b>CreationDate</b>	date	Date and time the card was created; always has a value
<b>PrintedDate</b>	date	Date and time the card was printed; may be blank
<b>Used</b>	true/false	Has the card been used yet?
<b>Value</b>	decimal	Decimal representation of the dollar value for gift cards and discount percent for discount cards. Always returns 0 for charge accounts.
<b>ValueUsed</b>	decimal	Decimal representation of the value used so far for gift cards and charge accounts. Always returns 0 for discount cards.
<b>FirstName</b>	string(25)	Optional value representing the card holder's first name
<b>LastName</b>	string(30)	Optional value representing the card holder's last name
<b>Text1</b>	string(50)	Optional value for a free text field to display on the card
<b>Text2</b>	string(50)	Optional value for a free text field to display on the card
<b>Text3</b>	string(50)	Optional value for a free text field to display on the card

If there is activity for the card, an Activity section made up of summary ActivityEntry items with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>ActivityID</b>	int	
<b>ActivityTypeID</b>	int	ID number of the activity type
<b>ActivityTypeName</b>	string	Friendly name of the activity type
<b>Date</b>	date	
<b>Value</b>	decimal	Decimal representation of the value used or changed
<b>Operator</b>	string(50)	An optional, user-supplied field to identify who performed the activity,

		such as the person using a cash register
<b>Location</b>	string(50)	An optional, user-supplied field to identify where the activity was performed, such as the cash register number

### Sample Call

This API call would request the details for the card GIFT1234 as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gccard&barcode=GIFT1234](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gccard&barcode=GIFT1234)

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<CardInformation>
  <CardID>27063296-32f8-4bd1-865c-2b1f3c32fed8</CardID>
  <Barcode>GIFT1234</Barcode>
  <TypeID>0</TypeID>
  <TypeName>Gift Card</TypeName>
  <TemplateID>2</TemplateID>
  <TemplateName>Gift - $50</TemplateName>
  <SingleUse>>false</SingleUse>
  <CreationDate>2018-05-31T10:00:00</Date>
  <PrintedDate>2018-05-31T10:00:00</Date>
  <Used>>true</Used>
  <Value>50.0000</Value>
  <ValueUsed>5.0000</ValueUsed>
  <FirstName>Import</FirstName>
  <LastName>Test</LastName>
  <Text1/>
  <Text2/>
  <Text3/>
  <Activity>
    <ActivityEntry>
      <ActivityID>27</ActivityID>
      <ActivityTypeID>2</ActivityTypeID>
      <ActivityTypeName>Created</ActivityTypeName>
      <Date>2018-05-17T22:59:37.957</Date>
      <Value>50.0000</Value>
      <Operator>John</Operator>
      <Location>Web</Location>
    </ActivityEntry>
    <ActivityEntry>
      <ActivityID>35</ActivityID>
      <ActivityTypeID>0</ActivityTypeID>
      <ActivityTypeName>Value Used</ActivityTypeName>
      <Date>2018-05-31T11:58:10.667</Date>
      <Value>5.0000</Value>
      <Operator>2</Operator>
      <Location>POS45</Location>
    </ActivityEntry>
  </Activity>
</CardInformation>
```

```
</ActivityEntry>
</Activity>
</CardInformation>
```

## gcdelete: Delete Gift Card

This call will permanently deactivate a gift card, preventing it from being used again. All existing activity will be saved and a new activity entry for the deletion will be created.

### Parameters

The **action** parameter must be set to **gcdelete** and you must specify either the **barcode** or **id** parameter.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gcdelete	<b>Required</b>
<b>barcode</b>	string(50)	0 – 9, a – z, A – Z	<b>Must specify this or the id parameter</b>
<b>id</b>	GUID		<b>Must specify this or the barcode parameter</b>

### Response

This call returns a [Generic Action Response](#). See the linked section for more details.

### Sample Call

This API call would delete the card GIFT1234 and request an XML response:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcdelete&barcode=GIFT1234
```

### Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<Result>
  <Status>OK</Status>
  <Message />
</Result>
```

## gcdeleteusage: Delete Gift Card Usage

This call erases all usage activity for a gift card, returning it to the full value and allowing it to be used again.

### Parameters

The **action** parameter must be set to **gcdeleteusage** and you must specify either the **barcode** or **id** parameter.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gcdeleteusage	<b>Required</b>

<b>barcode</b>	string(50)	0 – 9, a – z, A – Z	<b>Must specify this or the id parameter</b>
<b>id</b>	GUID		<b>Must specify this or the barcode parameter</b>

## Response

This call returns a [Generic Action Response](#). See the linked section for more details.

## Sample Call

This API call would delete the usage for card GIFT1234 and request an XML response:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcdeleteusage&barcode=GIFT1234
```

## Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<Result>
  <Status>OK</Status>
  <Message />
</Result>
```

## gcsearch: Search Gift Cards

This call returns a list of gift card details based on the specified search criteria.

## Parameters

The **action** parameter must be set to **gcsearch**. All search parameters are optional.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gcsearch	<b>Required</b>
<b>template</b>	int		ID number of the <a href="#">template</a>
<b>type</b>	int	0 – 2	ID number of the <a href="#">card type</a>
<b>first</b>	string(25)		First name on the card
<b>last</b>	string(30)		Last name on the card
<b>text</b>	string(50)		Any of the three text fields on the card
<b>hideprinted</b>	true/false		Hide cards that have been printed

## Response

The response will contain a list of CardInformation objects that match the search criteria, which you can find detailed in the **gccard** section above.

### Sample Call

This API call would search for all cards with the template ID 2 and request a response as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcsearch&template=2
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SearchResults>
  <CardInformation>See the gccard section above for details.</CardInformation>
</SeachResults>
```

## gctemplates: Get Gift Card Templates

This call will return a list of ID numbers and other information about the templates for gift and discount cards and charge accounts. All of the default values can be changed when generating cards, so make sure to use the **gccard** call to get the specific card's values.

### Parameters

The **action** parameter must be set to **gctemplates**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gctemplates	<b>Required</b>

### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>TemplateID</b>	int	
<b>Name</b>	string(50)	
<b>BarcodePrefix</b>	string(10)	The barcode prefix to use when automatically generating barcodes. Can be overridden with various add/import methods.
<b>BarcodeRandomCharacters</b>	int	The number of random characters to append when automatically generating barcodes. Can be overridden with various add/import methods.
<b>DefaultTypeID</b>	int	The ID of the default type (gift, discount, charge) for this template, but can be changed when generating cards.

<b>DefaultTypeName</b>	string(50)	The name of the default type for this template, but can be changed when generating cards.
<b>DefaultValue</b>	decimal	The default value (gift card dollar amount or discount percent) for this template, but can be changed when generating cards.
<b>DefaultSingleUse</b>	true/false	Whether the card can be used one or more times for this template by default, but can be changed when generating cards.
<b>PrintedCardXAML</b>	string	The XAML code used to render the card for printing on a card printer.
<b>PrintedCardBackgroundURL</b>	string	The URL of the printed card's background image on our Amazon S3 bucket.
<b>ElectronicCardHTML</b>	string	The HTML code used to render the card in a web browser.

### Sample Call

This API call would request the list of templates as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gctemplates](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gctemplates)

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Templates>
  <Template>
    <TemplateID>2</TemplateID>
    <Name>Gift - $50</Name>
    <BarcodePrefix>G50</BarcodePrefix>
    <BarcodeRandomCharacters>4</BarcodeRandomCharacters>
    <DefaultTypeID>0</DefaultTypeID>
    <DefaultTypeName>Gift Card</DefaultTypeName>
    <DefaultValue>50.0000</DefaultValue>
    <DefaultSingleUse>>false</DefaultSingleUse>
    <PrintedCardXAML>XAML Code</PrintedCardXAML>
    <PrintedCardBackgroundURL>Image URL</PrintedCardBackgroundURL>
    <ElectronicCardHTML>HTML Code</ElectronicCardHTML>
  </Template>
</Templates>
```

## gctypes: Get Gift Card Types

This call will return a list of ID numbers and text explanations of the types of cards in the gift card system. This information is also included automatically with the **gccard** API call.

### Parameters

The **action** parameter must be set to **gctypes**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gctypes	<b>Required</b>

### Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>ID</b>	int	
<b>Name</b>	string	

### Sample Call

This API call would request the list of card types as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gctypes
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Types>
  <Type>
    <ID>0</ID>
    <Name>Gift Card</Name>
  </Type>
  <Type>
    <ID>1</ID>
    <Name>Discount Card</Name>
  </Type>
  <Type>
    <ID>2</ID>
    <Name>Charge Account</Name>
  </Type>
</Types>
```

### Current Values

This API call will always return these results. If we add more card types in the future, we will add on to the end and not change any existing values, and the API call and guide will be updated.

ID	Name	Notes
----	------	-------

0	Gift Card	Can use the dollar amount up to the specified value; single use if specified
1	Discount Card	Can receive the specified discount value; single use if specified
2	Charge Account	Can accept unlimited transactions for any dollar amount

## gcupdate: Update Gift Card Details

This call updates a card's details. You must specify at least one parameter to update or the call will fail.

### Parameters

The **action** parameter must be set to **gcupdate** and you must specify either the **barcode** or **id** parameter, along with at least one parameter to update.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gcupdate	<b>Required</b>
<b>barcode</b>	string(50)	0 – 9, a – z, A – Z	<b>Must specify this or the id parameter</b>
<b>id</b>	GUID		<b>Must specify this or the barcode parameter</b>
<b>template</b>	int		ID number of the <a href="#">template</a>
<b>type</b>	int	0 – 2	ID number of the <a href="#">card type</a>
<b>value</b>	decimal		Dollar value for gift cards, discount amount for discount cards
<b>singleuse</b>	true/false		Whether the card can be used more than once
<b>first</b>	string(25)		First name on the card
<b>last</b>	string(30)		Last name on the card
<b>text1</b>	string(50)		First text field
<b>text2</b>	string(50)		Second text field
<b>text3</b>	string(50)		Third text field
<b>printdate</b>	string	set, clear	<b>set</b> automatically uses the current date

### Response

This call returns a [Generic Action Response](#). See the linked section for more details.



## Sample Call

This API call would clear the print date for card GIFT1234 and request an XML response:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcupdate&barcode=GIFT1234&printdate=clear
```

## Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<Result>
  <Status>OK</Status>
  <Message />
</Result>
```

## gcuse: Use Card

This call allows you to use a card. The safety check parameter below will automatically enforce a card's single use property or maximum value and can be disabled to force a transaction through that would normally not be allowed. An activity entry will be created automatically upon a successful transaction.

## Parameters

The **action** parameter must be set to **gcuse** and there are several mandatory and optional parameters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	gccard	<b>Required</b>
<b>barcode</b>	string(50)	0 – 9, a – z, A – Z	<b>Required</b>
<b>value</b>	decimal		<b>Required</b> for gift cards and charge accounts; ignored for discount cards
<b>operator</b>	string(50)		<b>Optional</b>
<b>location</b>	string(50)		<b>Optional</b>
<b>safety</b>	true/false		<b>Optional</b> but defaults to true if not specified

## Response

A successful transaction will return the **ActivityID** value assigned to the record:

Value	Data Type	Notes
<b>ActivityID</b>	int	

A missing parameter or failed safety check will return an HTTP 400 status with a status description explaining the error, such as "Failed safety check: already used" and no activity entry will be created.

### Sample Call

This API call would use \$5 on the gift card GIFT1234 and get a response as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=gcuse&barcode=GIFT12345&value=5
```

Since no operator and location were specified those fields will be blank, but since the safety parameter was omitted the safety check will be performed by default.

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<ActivityID>36</ActivityID>
```

## API Calls: Ticketing

This section covers all API calls for the ticket sales module.

### otsorder: Get Ticket Order

This call will return order details, order items, barcode details and scanning records for a single ticket order.

#### Parameters

The **action** parameter must be set to **otsorder** and you must specify the **id** parameter.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	otsorder	<b>Required</b>
<b>id</b>	GUID	Any valid order number	The system's globally unique order number

#### Response

The response will include the following values:

Value	Data Type	Notes
<b>ID</b>	GUID	The system's globally unique order number
<b>OrderType</b>	string	
<b>OrderDate</b>	date	
<b>LandingPageID</b>	GUID	Filled in if placed on the non-default landing page
<b>LandingPageName</b>	string(50)	Filled in if placed on the non-default landing page
<b>DeliveryMethod</b>	string	
<b>PaymentType</b>	string	
<b>ProcessorTransactionID</b>	string	
<b>CreditCardLastFour</b>	string	Filled in if credit card sale
<b>CreditCardAuthCode</b>	string	Filled in if credit card sale
<b>CheckNumber</b>	string	Filled in if check sale
<b>CheckName</b>	string	Filled in if check sale
<b>Subtotal</b>	decimal	
<b>SalesTax</b>	decimal	
<b>ProcessingFee</b>	decimal	
<b>Delivery</b>	decimal	

---

<b>Donation</b>	decimal	
<b>DiscountID</b>	GUID	Filled in if discount used
<b>DiscountName</b>	string(50)	Filled in if discount used
<b>DiscountAmount</b>	decimal	Filled in if discount used
<b>CharityID</b>	GUID	Fill in if charitable order
<b>CharityName</b>	string(50)	Fill in if charitable order
<b>CharityRepresentative</b>	string(50)	Fill in if charitable order and representative name supplied
<b>CharityAmount</b>	decimal	Fill in if charitable order
<b>TradeCategoryID</b>	int	Fill in if trade or donation category applied to order
<b>TradeCategoryName</b>	string(50)	Fill in if trade or donation category applied to order
<b>BillingFirstName</b>	string(25)	
<b>BillingLastName</b>	string(35)	
<b>BillingAddress1</b>	string(50)	
<b>BillingAddress2</b>	string(50)	
<b>BillingCity</b>	string(25)	
<b>BillingStateProvince</b>	string(25)	
<b>BillingCountry</b>	string(25)	
<b>BillingZipCode</b>	string(15)	
<b>BillingPhone</b>	string(15)	
<b>BillingEMail</b>	string(50)	
<b>ShippingFirstName</b>	string(25)	
<b>ShippingLastName</b>	string(35)	
<b>ShippingAddress1</b>	string(50)	
<b>ShippingAddress2</b>	string(50)	
<b>ShippingCity</b>	string(25)	
<b>ShippingStateProvince</b>	string(25)	
<b>ShippingCountry</b>	string(25)	
<b>ShippingZipCode</b>	string(15)	
<b>IPAddress</b>	string(15)	
<b>Referrer</b>	string(50)	Result of the “how did you hear about this event?” prompt
<b>AdNetworkCode</b>	string(15)	Result of the adnet query string parameter passed to a landing page

---

<b>UTMCampaign</b>	string(250)	Result of the utm_campaign query string parameter passed to a landing page.
<b>UTMContent</b>	string(250)	Result of the utm_content query string parameter passed to a landing page.
<b>UTMMedium</b>	string(250)	Result of the utm_medium query string parameter passed to a landing page.
<b>UTMSource</b>	string(250)	Result of the utm_source query string parameter passed to a landing page.
<b>UTMTerm</b>	string(250)	Result of the utm_term query string parameter passed to a landing page.
<b>GoogleAdsSearchID</b>	string(250)	Result of the Google Ads Search ID query string parameter passed to a landing page.
<b>GoogleAdsDisplayID</b>	string(250)	Result of the Google Ads Display ID query string parameter passed to a landing page.
<b>GoogleAnalyticsSessionID</b>	string(250)	Result of the Google Analytics Session ID query string parameter or cookie passed to a landing page.
<b>Comments</b>	string(500)	
<b>PastPurchaser</b>	true/false	

If there are tickets on the order, a Tickets section made up of summary Ticket items with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>ID</b>	GUID	The system's globally unique ticket ID
<b>Name</b>	string(50)	The ticket's internal name
<b>ReportingGroup</b>	string(25)	An optional, user-supplied field that groups items on certain reports
<b>LineItemID</b>	string(30)	An optional, user-supplied field that contains a custom ID number displayed on reports and passed to the credit card processor
<b>Quantity</b>	int	The quantity of this type of ticket on the order
<b>QuantityFree</b>	int	The quantity of this type of ticket that was added through free ticket or buy X get 1 discounts.

<b>Price</b>	decimal	The price for this item at the time of sale.
<b>ProcessingFee</b>	decimal	The processing fee for this item at the time of sale.
<b>SalesTax</b>	decimal	The sales tax for this item at the time of sale.

If there are packages on the order, a Packages section made up of summary Package items with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>ID</b>	GUID	The system's globally unique package ID
<b>Name</b>	string(50)	The package's internal name
<b>ReportingGroup</b>	string(25)	An optional, user-supplied field that groups items on certain reports
<b>LineItemID</b>	string(30)	An optional, user-supplied field that contains a custom ID number displayed on reports and passed to the credit card processor
<b>Quantity</b>	int	The quantity of this type of package on the order
<b>QuantityFree</b>	int	The quantity of this type of ticket that was added through free ticket or buy X get 1 discounts.
<b>Price</b>	decimal	The price for this item at the time of sale.
<b>ProcessingFee</b>	decimal	The processing fee for this item at the time of sale.
<b>SalesTax</b>	decimal	The sales tax for this item at the time of sale.

If the items on an order generate barcodes or have seating information, an IssuedTickets section with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>ID</b>	GUID	The system's globally unique ticket ID
<b>Name</b>	string(50)	The ticket's internal name
<b>Barcode</b>	string(50)	
<b>SeatArea</b>	string(50)	The seating area's internal name

<b>SeatRow</b>	string(5)
<b>SeatCol</b>	string(5)

If there is any scanning activity for the order, a ScanRecords section with the following information will be added:

Value	Data Type	Notes
<b>ActivityID</b>	int	Unique, sequential ID per scan
<b>Date</b>	date	Includes time and time zone
<b>Barcode</b>	string(20)	
<b>TicketName</b>	string(50)	
<b>LocationID</b>	int	
<b>LocationName</b>	string(50)	
<b>LocationIsEntrance</b>	true/false	Whether the location's scans count towards total attendance
<b>ResultCode</b>	int	
<b>ResultMessage</b>	string	

**Note:** This ScanRecords section does not have the TicketInformation or CredentialInformation subsections that a call to **acactivity** would, since you already have the ticket information in the order details.

### Sample Call

This API call would request details for order **61862043-5ba7-4810-80a8-8b7a0d506353** as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otsorder&id=61862043-5ba7-4810-80a8-8b7a0d506353](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otsorder&id=61862043-5ba7-4810-80a8-8b7a0d506353)

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Order>
  <ID>61862043-5ba7-4810-80a8-8b7a0d506353</ID>
  <OrderType>Online</OrderType>
  <OrderDate>2016-07-29T20:00:00.00</OrderDate>
  <LandingPageID/>
  <LandingPageName/>
  <DeliveryMethod>Print Online</DeliveryMethod>
  <PaymentType>Visa</PaymentType>
  <ProcessorTransactionID>1234567890</ProcessorTransactionID>
  <CreditCardLastFour>1234</CreditCardLastFour>
  <CreditCardAuthCode>123456</CreditCardAuthCode>
  <CheckNumber/>
  <CheckName/>
  <Subtotal>19.8000</Subtotal>
```

<SalesTax>0.0000</SalesTax>  
<ProcessingFee>1.8000</ProcessingFee>  
<Delivery>0.0000</Delivery>  
<Donation>0.0000</Donation>  
<DiscountID>14c68307-b786-44ef-8a56-fadf365dfb39</DiscountID>  
<DiscountName>10% Off</DiscountName>  
<DiscountAmount>-2.0000</DiscountAmount>  
<CharityID/>  
<CharityName/>  
<CharityRepresentative/>  
<CharityAmount>0.0000</CharityAmount>  
<TradeCategoryID/>  
<TradeCategoryName/>  
<BillingFirstName>Cassie</BillingFirstName>  
<BillingLastName>Cueto</BillingLastName>  
<BillingAddress1>123 North Street</BillingAddress1>  
<BillingAddress2>Appt 514</BillingAddress2>  
<BillingCity>Hagerstown</BillingCity>  
<BillingStateProvince>MD</BillingStateProvince>  
<BillingCountry>United States</BillingCountry>  
<BillingZipCode>21740</BillingZipCode>  
<BillingPhone>1231231234</BillingPhone>  
<BillingEMail>cassie@cuetoems.com</BillingEMail>  
<ShippingFirstName/>  
<ShippingLastName/>  
<ShippingAddress1/>  
<ShippingAddress2/>  
<ShippingCity/>  
<ShippingStateProvince/>  
<ShippingCountry/>  
<ShippingZipCode/>  
<IPAddress>192.168.1.1</IPAddress>  
<Referrer>Other</Referrer>  
<AdNetworkCode/>  
<UTMCampaign>A71BC89A LAUNCH</UTMCampaign>  
<UTMContent />  
<UTMMedium>email</UTMMedium>  
<UTMSource>Sales Launch Blast</UTMSource>  
<UTMTerm />  
<GoogleAdsSearchID />  
  
<GoogleAdsDisplayID />  
<GoogleAnalyticsSessionID />  
<Comments/>  
<PastPurchaser>>false</PastPurchaser>  
<Tickets>  
  <Ticket>  
    <ID>350d7bd5-625c-4045-a9b2-d4e6d3079c76</ID>  
    <Name>Any Day Grounds</Name>  
    <ReportingGroup>Grounds</ReportingGroup>  
    <LineItemID>ADG</LineItemID>  
    <Quantity>2</Quantity>  
    <QuantityFree>0</QuantityFree>  
    <Price>9.0000</Price>  
    <ProcessingFee>0.9000</ProcessingFee>



```

    <SalesTax>0.0000</SalesTax>
  </Ticket>
</Tickets>
<Packages/>
<IssuedTickets>
  <IssuedTicket>
    <ID>350d7bd5-625c-4045-a9b2-d4e6d3079c76</ID>
    <Name>Any Day Grounds</Name>
    <Barcode>TQG4P0R3</Barcode>
    <SeatArea/>
    <SeatRow/>
    <SeatCol/>
  </IssuedTicket>
</IssuedTickets>
<ScanRecords>
  <ScanRecord>
    <ActivityID>36</ActivityID>
    <Date>2017-11-20T15:32:11.693</Date>
    <Barcode>TL9CUY2W</Barcode>
    <TicketName>Any Day Grounds</TicketName>
    <LocationID>2</LocationID>
    <LocationName>Front Gate</LocationName>
    <LocationIsEntrance>true</LocationIsEntrance>
    <ResultCode>1</ResultCode>
    <ResultMessage>Allowed - Credential</ResultMessage>
  </ScanRecord>
</ScanRecords>
</Order>

```

## otsorders: Get Ticket Orders

This call will return a list of ID numbers, order details, order items, barcode details and scanning records for each ticket order in the system that matches your filter criteria.

### Parameters

The **action** parameter must be set to **otsorders** and you may specify some optional filters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	otsorders	<b>Required</b>
<b>mindate</b>	date	Any date/time format	<b>Optional</b> - Specifies a minimum order date/time for records
<b>maxdate</b>	date	Any date/time format	<b>Optional</b> - Specifies a maximum order date/time for records

### Response

The response will contain a list of Order objects, which you can find detailed in the **otsorder** section above.

## Sample Call

This API call would request the list of orders from July 1, 2016 to July 31, 2016 as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otsorders&mindate=2016-07-01&maxdate=2016-07-31
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Orders>
  <Order>See the otsorder section above for details.</Order>
</Orders>
```

## otspackages: Get Ticket Packages

This call will return a list of ID numbers, names, properties and settings for each ticket in the system. Packages may contain any number of tickets.

### Parameters

The **action** parameter must be set to **otspackages**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	otspackages	<b>Required</b>

### Response

The response will include the following values for each package:

Value	Data Type	Notes
<b>ID</b>	GUID	
<b>Name</b>	string(50)	The internal name, shown to staff and on reports
<b>DisplayName</b>	string(100)	The public name, shown to the public
<b>Description</b>	string	May contain user-supplied HTML. <b>Do not display without sanitizing.</b>
<b>Price</b>	decimal	
<b>ProcessingFee</b>	decimal	
<b>ProcessingFeeTerm</b>	string(20)	Used to customize the term for processing fees on the checkout page
<b>SalesTax</b>	decimal	
<b>ShowForPublicSales</b>	true/false	Whether the item is visible on the default public ticket sales page
<b>ShowForStaffSales</b>	true/false	Whether the item is visible on the back end staff order tool

<b>ShowForWillCall</b>	true/false	Whether the item is visible on the will call tools
<b>Charitable</b>	true/false	If a charitable item is on an order, the system will show the charity selection interface
<b>DisplayGroup</b>	string(100)	The name of the display group that the package is listed under on the sales page
<b>ReportingGroup</b>	string(25)	An optional, user-supplied field that groups items on certain reports
<b>LineItemID</b>	string(30)	An optional, user-supplied field that contains a custom ID number displayed on reports and passed to the credit card processor

If the package contains tickets, a Tickets section made up of summary Ticket items with the following information will be added:

Value	Data Type	Notes
<b>ID</b>	GUID	The system's globally unique ticket ID
<b>Name</b>	string(50)	The ticket's internal name
<b>ReportingGroup</b>	string(25)	An optional, user-supplied field that groups items on certain reports
<b>LineItemID</b>	string(30)	An optional, user-supplied field that contains a custom ID number displayed on reports and passed to the credit card processor
<b>Quantity</b>	int	The quantity of this type of ticket in the package

### Sample Call

This API call would request the list of packages as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otspackages](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otspackages)

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Packages>
  <Package>
    <ID>923268a9-ed69-41ef-9732-5e4053f25b3f</ID>
    <Name>Weekly Grounds</Name>
    <DisplayName>Weekly Grounds</DisplayName>
```

```

<Description>One grounds ticket for each day of the week.</Description>
<Price>50.0000</Price>
<ProcessingFee>1.0000</ProcessingFee>
<ProcessingFeeTerm>Processing Fee</ProcessingFeeTerm>
<SalesTax>0.0000</SalesTax>
<ShowForPublicSales>>true</ShowForPublicSales>
<ShowForStaffSales>>true</ShowForStaffSales>
<ShowForWillCall>>false</ShowForWillCall>
<Charitable>>false</Charitable>
<DisplayGroup>Grounds Tickets</DisplayGroup>
<ReportingGroup>Grounds</ReportingGroup>
<LineItemID>WG</LineItemID>
<Tickets>
  <Ticket>
    <ID>350d7bd5-625c-4045-a9b2-d4e6d3079c76</ID>
    <Name>Any Day Grounds</Name>
    <ReportingGroup>Grounds</ReportingGroup>
    <LineItemID>ADG</LineItemID>
    <Quantity>7</Quantity>
  </Ticket>
</Tickets>
</Package>
</Packages>

```

## otstickets: Get Tickets

This call will return a list of ID numbers, names, properties and settings for each ticket in the system. Tickets are single-page items.

### Parameters

The **action** parameter must be set to **otstickets**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	otstickets	<b>Required</b>

### Response

The response will include the following values for each ticket:

Value	Data Type	Notes
<b>ID</b>	GUID	
<b>Name</b>	string(50)	The internal name, shown to staff and on reports
<b>DisplayName</b>	string(100)	The public name, shown to the public
<b>Description</b>	string	May contain user-supplied HTML. <b>Do not display without sanitizing.</b>
<b>Date</b>	date	Includes time component but will always be set to midnight

---

<b>Price</b>	decimal	
<b>ProcessingFee</b>	decimal	
<b>ProcessingFeeTerm</b>	string(20)	Used to customize the term for processing fees on the checkout page
<b>SalesTax</b>	decimal	
<b>ShowForPublicSales</b>	true/false	Whether the item is visible on the default public ticket sales page
<b>ShowForStaffSales</b>	true/false	Whether the item is visible on the back end staff order tool
<b>ShowForWillCall</b>	true/false	Whether the item is visible on the will call tools
<b>HasETicketTemplate</b>	true/false	Whether the item has an e-ticket template
<b>MobileTicket</b>	true/false	If mobile tickets are enabled, show this item in the mobile ticket interface
<b>Charitable</b>	true/false	If a charitable item is on an order, the system will show the charity selection interface
<b>DisplayGroup</b>	string(100)	The name of the display group that the ticket is listed under on the sales page
<b>ScanMode</b>	int	
<b>ScanModeName</b>	string	
<b>ReportingGroup</b>	string(25)	An optional, user-supplied field that groups items on certain reports
<b>LineItemID</b>	string(30)	An optional, user-supplied field that contains a custom ID number displayed on reports and passed to the credit card processor

---

### Sample Call

This API call would request the list of tickets as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otstickets](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=otstickets)

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Tickets>
  <Ticket>
    <ID>350d7bd5-625c-4045-a9b2-d4e6d3079c76</ID>
    <Name>Any Day Grounds</Name>
    <DisplayName>Any Day Grounds</DisplayName>
```

```
<Description>Grounds access any single day of the event.</Description>
<Date>2018-07-29T00:00:00</Date>
<Price>10.0000</Price>
<ProcessingFee>1.0000</ProcessingFee>
<ProcessingFeeTerm>Service Charge</ProcessingFeeTerm>
<SalesTax>0.0000</SalesTax>
<ShowForPublicSales>true</ShowForPublicSales>
<ShowForStaffSales>true</ShowForStaffSales>
<ShowForWillCall>true</ShowForWillCall>
<HasETicketTemplate>true</HasETicketTemplate>
<MobileTicket>true</MobileTicket>
<Charitable>false</Charitable>
<DisplayGroup>Tickets</DisplayGroup>
<ScanMode>1</ScanMode>
<ScanModeName>Single Use</ScanModeName>
<ReportingGroup>Grounds</ReportingGroup>
<LineItemID>ADG</LineItemID>
</Ticket>
</Tickets>
```

## API Calls: Volunteers

This section covers all API calls for the volunteer management module.

### volcheckin: Volunteer Shift Check In or Out

This call allows you to check a volunteer in or out of a shift assignment.

#### Parameters

The **action** parameter must be set to **volcheckin** and you must specify the **id** and **status** parameters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	volphotoproblem	<b>Required</b>
<b>id</b>	int	Any valid shift assignment ID	Get the assignment ID by looking at the first field for each entry in the Shifts section of a volunteer's profile
<b>status</b>	true/false		Set to <b>true</b> to check in or <b>false</b> to check out

#### Response

This call returns a [Generic Action Response](#). See the linked section for more details.

#### Sample Call

This API call would perform a check in for the assignment ID **154** and request an XML response:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volcheckin&id=154&status=true
```

#### Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<Result>
  <Status>OK</Status>
  <Message />
</Result>
```

### volcommittees: Get Committees and Member Counts

This call will return a list of ID numbers, names and counts for each committee in the system.

#### Parameters

The **action** parameter must be set to **volcommittees**. There are no optional values.

Parameter	Data Type	Possible Values	Notes
-----------	-----------	-----------------	-------

<b>action</b>	string	volcommittees	<b>Required</b>
---------------	--------	---------------	-----------------

## Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>CommitteeID</b>	int	
<b>Name</b>	string(50)	
<b>MembersDesired</b>	int	
<b>MembersAssigned</b>	int	How many volunteer accounts are assigned to the committee
<b>ChairsAssigned</b>	int	How many chair accounts are assigned to the committee

## Sample Call

This API call would request the list of committees as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volcommittees
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Committees>
  <Committee>
    <CommitteeID>36</CommitteeID>
    <Name>Admissions</Name>
    <MembersDesired>40</MembersDesired>
    <MembersAssigned>33</MembersAssigned>
    <ChairsAssigned>2</ChairsAssigned>
  </Committee>
</Committees>
```

## volpayments: Get Volunteer Payments

This call will return transaction details for each volunteer payment that matches the specified criteria.

### Parameters

The **action** parameter must be set to **volpayments** and you may specify some optional filters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	volpayments	<b>Required</b>
<b>mindate</b>	date	Any date/time format	<b>Optional</b> - Specifies a minimum date/time for records



<b>maxdate</b>	date	Any date/time format	<b>Optional</b> - Specifies a maximum date/time for records
----------------	------	----------------------	---

## Response

The response will include the following values for each record:

Value	Data Type	Notes
<b>UserID</b>	GUID	The system's globally unique user ID
<b>FirstName</b>	string(25)	
<b>LastName</b>	string(35)	
<b>PaymentID</b>	GUID	The system's globally unique payment ID
<b>Date</b>	date	
<b>PaymentType</b>	string	
<b>Subtotal</b>	decimal	
<b>SalesTax</b>	decimal	
<b>ProcessorTransactionID</b>	string	
<b>CreditCardLastFour</b>	string	Filled in if credit card sale
<b>CreditCardAuthCode</b>	string	Filled in if credit card sale
<b>CheckNumber</b>	string	Filled in if check sale
<b>CheckName</b>	string	Filled in if check sale
<b>IPAddress</b>	string(15)	

## Sample Call

This API call would request the list of payments between July 1, 2016 and July 31, 2016 as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volpayments&mindate=2016-07-01&maxdate=2016-07-31](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volpayments&mindate=2016-07-01&maxdate=2016-07-31)

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Payments>
  <Payment>
    <UserID>49809c5f-ffb0-4598-a758-4e5b96fae9bb</UserID>
    <FirstName>Cassie</FirstName>
    <LastName>Cueto</LastName>
    <PaymentID>1652efd3-829d-4bbf-8a73-05abf86d0f99</PaymentID>
    <Date>2016-07-29T13:45:13.133</Date>
    <PaymentType>Visa</PaymentType>
    <Subtotal>95.0000</Subtotal>
  </Payment>
</Payments>
```

```

    <SalesTax>21.0000</SalesTax>
    <ProcessorTransactionID>1234567890</ProcessorTransactionID>
    <CreditCardLastFour>1234</CreditCardLastFour>
    <CreditCardAuthCode>123456</CreditCardAuthCode>
    <CheckNumber/>
    <CheckName/>
    <IPAddress>192.168.1.1</IPAddress>
  </Payment>
</Payments>

```

## volphotoproblem: Flag/Unflag Bad Photograph

This call allows you to flag, or cancel the flagging, of a volunteer profile that has uploaded a photograph that doesn't meet your quality standards.

### Parameters

The **action** parameter must be set to **volphotoproblem** and you must specify the **id** and **problem** parameters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	volphotoproblem	<b>Required</b>
<b>id</b>	GUID	Any valid user ID	The system's globally unique user ID
<b>problem</b>	true/false		Set to <b>true</b> to indicate a problem, <b>false</b> for no problem

### Response

This call returns a [Generic Action Response](#). See the linked section for more details.

### Sample Call

This API call would update the profile for the user ID **49809c5f-ffb0-4598-a758-4e5b96fae9bb** with a photo problem and request an XML response:

```

https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volphotoproblem&id=49809c5f-ffb0-4598-a758-4e5b96fae9bb&problem=true

```

### Sample Response

```

<?xml version="1.0" encoding="utf-8"?>
<Result>
  <Status>OK</Status>
  <Message />
</Result>

```

## volprofile: Get Volunteer Profile

This call will return details, custom question responses, committee information, uniform information, payments and schedule information for the specified user.

### Parameters

The **action** parameter must be set to **volprofile** and you must specify the **id** parameter.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	volprofile	<b>Required</b>
<b>id</b>	GUID	Any valid user ID	The system's globally unique user ID

### Response

The response will include the following values:

Value	Data Type	Notes
<b>UserID</b>	GUID	The system's globally unique user ID
<b>Username</b>	string(50)	
<b>AccountType</b>	string(50)	Value will be Volunteer or Chairperson
<b>RegistrationStatus</b>	string	
<b>RegistrationDate</b>	date	
<b>RegistrationCodeID</b>	int	Filled in if the volunteer used a registration code
<b>RegistrationCodeName</b>	string(50)	Filled in if the volunteer used a registration code
<b>Agreement1Timestamp</b>	date	Filled in if the volunteer agreed to the first agreement prompt
<b>Agreement2Timestamp</b>	date	Filled in if the volunteer agreed to the second agreement prompt
<b>FirstName</b>	string(25)	
<b>MiddleName</b>	string(25)	
<b>LastName</b>	string(35)	
<b>Gender</b>	string(1)	
<b>Birthdate</b>	date	
<b>Address1</b>	string(50)	
<b>Address2</b>	string(50)	
<b>City</b>	string(25)	
<b>StateProvince</b>	string(25)	

---

<b>Country</b>	string(25)	
<b>ZipCode</b>	string(15)	
<b>ShippingAddress1</b>	string(50)	
<b>ShippingAddress2</b>	string(50)	
<b>ShippingCity</b>	string(25)	
<b>ShippingStateProvince</b>	string(25)	
<b>ShippingCountry</b>	string(25)	
<b>ShippingZipCode</b>	string(15)	
<b>HomePhone</b>	string(15)	
<b>WorkPhone</b>	string(15)	
<b>CellPhone</b>	string(15)	
<b>EMail</b>	string(50)	
<b>EMailVerified</b>	true/false	
<b>DriversLicense</b>	true/false	
<b>EmergencyContactName</b>	string(50)	
<b>EmergencyContactRelationship</b>	string(50)	
<b>EmergencyContactPhone</b>	string(15)	
<b>GuardianContactName</b>	string(50)	Prompt displayed if volunteer under 18 years old
<b>GuardianContactRelationship</b>	string(50)	Prompt displayed if volunteer under 18 years old
<b>GuardianContactEMail</b>	string(50)	Prompt displayed if volunteer under 18 years old
<b>YearsOfService</b>	int	
<b>PreviousExperience</b>	string	
<b>VolunteerWith</b>	string(50)	
<b>CanHelpBeforeEvent</b>	true/false	
<b>CanHelpAfterEvent</b>	true/false	
<b>CanWorkMultiShiftDay</b>	true/false	
<b>PaymentStatus</b>	string	
<b>UniformOrderReceived</b>	true/false	
<b>Notes</b>	string	
<b>PhotoURL</b>	string	Filled in if photo uploaded, otherwise blank
<b>PhotoProblem</b>	true/false	Has the photograph been flagged for replacement for a quality problem?

---

If a volunteer answered any custom questions, a Questions section with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>QuestionID</b>	int	
<b>ShortName</b>	string(25)	Friendly name used for reporting
<b>Text</b>	string	Full text of the question
<b>Response</b>	string	Full text of the response

If a volunteer selected committee preferences, a CommitteePreferences section with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>CommitteeID</b>	int	
<b>Name</b>	string(50)	

If the volunteer has been assigned to at least one committee, a CommitteeAssignments section with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>CommitteeID</b>	int	
<b>Name</b>	string(50)	
<b>AssignmentDate</b>	date	

If the volunteer has selected uniform items, a UniformItems section with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
<b>Name</b>	string(50)	
<b>SKU</b>	string(25)	
<b>Size</b>	string(25)	
<b>Color</b>	string(25)	
<b>Quantity</b>	int	

If the volunteer has submitted a payment, a Payments section with the following information will be added:

<b>Value</b>	<b>Data Type</b>	<b>Notes</b>
--------------	------------------	--------------

<b>PaymentID</b>	GUID	The system's globally unique payment ID
<b>Date</b>	date	
<b>PaymentType</b>	string	
<b>Subtotal</b>	decimal	
<b>SalesTax</b>	decimal	
<b>ProcessorTransactionID</b>	string	
<b>CreditCardLastFour</b>	string	Filled in if credit card sale
<b>CreditCardAuthCode</b>	string	Filled in if credit card sale
<b>CheckNumber</b>	string	Filled in if check sale
<b>CheckName</b>	string	Filled in if check sale
<b>IPAddress</b>	string(15)	

If the volunteer has been assigned to shifts, a Shifts section with the following information will be added:

Value	Data Type	Notes
<b>AssignmentID</b>	int	This assignment's unique ID number that can be passed to other API calls.
<b>CommitteeName</b>	string(50)	
<b>ShiftName</b>	string(50)	
<b>StartTime</b>	date	
<b>EndTime</b>	date	
<b>Location</b>	string(50)	
<b>Task</b>	string(50)	
<b>CheckedIn</b>	true/false	

### Sample Call

This API call would request the profile for the user ID **49809c5f-ffb0-4598-a758-4e5b96fae9bb** as XML:

[https://www.cuetoems.com/cems\\_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volprofile&id=49809c5f-ffb0-4598-a758-4e5b96fae9bb](https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volprofile&id=49809c5f-ffb0-4598-a758-4e5b96fae9bb)

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Volunteer>
  <UserID>49809c5f-ffb0-4598-a758-4e5b96fae9bb</UserID>
  <Username>cassie</Username>
  <RegistrationStatus>Registered</RegistrationStatus>
  <RegistrationDate>2016-07-29T13:34:26.393</RegistrationDate>
  <RegistrationCodeID/>
  <RegistrationCodeName/>
</Volunteer>
```

<Agreement1Timestamp>2016-07-29T13:34:26.393</Agreement1Timestamp>  
<FirstName>Cassie</FirstName>  
<MiddleName>Elgato</MiddleName>  
<LastName>Cueto</LastName>  
<Gender>F</Gender>  
<Birthdate>2000-07-29T00:00:00</Birthdate>  
<Address1>123 N Street</Address1>  
<Address2/>  
<City>Hagerstown</City>  
<StateProvince>MD</StateProvince>  
<Country>US</Country>  
<ZipCode>21740</ZipCode>  
<ShippingAddress1/>  
<ShippingAddress2/>  
<ShippingCity/>  
<ShippingStateProvince/>  
<ShippingCountry/>  
<ShippingZipCode/>  
<HomePhone>1231231234</HomePhone>  
<WorkPhone/>  
<CellPhone>1231231234</CellPhone>  
<EMail>cassie@cuetoems.com</EMail>  
<EMailVerified>>true</EMailVerified>  
<DriversLicense>>true</DriversLicense>  
<EmergencyContactName>John</EmergencyContactName>  
<EmergencyContactRelationship>Feeder</EmergencyContactRelationship>  
<EmergencyContactPhone>1231231234</EmergencyContactPhone>  
<GuardianContactName/>  
<GuardianContactRelationship/>  
<GuardianContactEMail/>  
<YearsOfService>3</YearsOfService>  
<PreviousExperience>I have worked at over 200 golf events.</PreviousExperience>  
<VolunteerWith/>  
<CanHelpBeforeEvent>>false</CanHelpBeforeEvent>  
<CanHelpAfterEvent>>true</CanHelpAfterEvent>  
<CanWorkMultiShiftDay>>true</CanWorkMultiShiftDay>  
<PaymentStatus>Paid</PaymentStatus>  
<UniformOrderReceived>>true</UniformOrderReceived>  
<Notes/>  
<PhotoURL/>  
<Questions>  
 <Question>  
 <QuestionID>4</QuestionID>  
 <ShortName>FavMusic</ShortName>  
 <Text>What is your favorite type of music?</Text>  
 <Response>Death Metal</Response>  
 </Question>  
</Questions>  
<CommitteePreferences>  
 <CommitteePreference>  
 <CommitteeID>43</CommitteeID>  
 <Name>Construction</Name>  
 </CommitteePreference>  
 <CommitteePreference>  
 <CommitteeID>36</CommitteeID>

```

    <Name>Admissions</Name>
  </CommitteePreference>
</CommitteePreferences>
<CommitteeAssignments>
  <Committee>
    <CommitteeID>36</CommitteeID>
    <Name>Admissions</Name>
    <AssignmentDate>2016-08-08T00:00:00</AssignmentDate>
  </Committee>
</CommitteeAssignments>
<UniformItems>
  <UniformItem>
    <Name>Women's Shirt</Name>
    <SKU>WSHIRT</SKU>
    <Size>XL</Size>
    <Color/>
    <Quantity>1</Quantity>
  </UniformItem>
</UniformItems>
<Payments>
  <Payment>
    <PaymentID>1652efd3-829d-4bbf-8a73-05abf86d0f99</PaymentID>
    <Date>2016-07-29T13:45:13.133</Date>
    <PaymentType>Visa</PaymentType>
    <Subtotal>95.0000</Subtotal>
    <SalesTax>21.0000</SalesTax>
    <ProcessorTransactionID>1234567890</ProcessorTransactionID>
    <CreditCardLastFour>1234</CreditCardLastFour>
    <CreditCardAuthCode>123456</CreditCardAuthCode>
    <CheckNumber/>
    <CheckName/>
    <IPAddress>192.168.1.1</IPAddress>
  </Payment>
</Payments>
<Shifts>
  <Shift>
    <AssignmentID>143</AssignmentID>
    <CommitteeName>Merchandise</CommitteeName>
    <ShiftName>Morning Shift</ShiftName>
    <StartTime>2017-08-01T06:00:00</StartTime>
    <EndTime>2017-08-01T11:30:00</EndTime>
    <Location/>
    <Task/>
    <CheckedIn>false</CheckedIn>
  </Shift>
</Shifts>
</Volunteer>

```

## volprofiles: Get Volunteer Profiles

This call will return a list of ID numbers, details, custom question responses, committee information, uniform information, payments and schedule information for each volunteer in the system that matches your filter criteria.



## Parameters

The **action** parameter must be set to **volprofiles** and you may specify some optional filters.

Parameter	Data Type	Possible Values	Notes
<b>action</b>	string	volprofiles	<b>Required</b>
<b>mindate</b>	date	Any date/time format	<b>Optional</b> - Specifies a minimum registration date/time for records
<b>maxdate</b>	date	Any date/time format	<b>Optional</b> - Specifies a maximum registration date/time for records

## Response

The response will contain a list of Volunteer objects, which you can find detailed in the **volprofile** section above.

## Sample Call

This API call would request the list of profiles for volunteers who registered between July 1, 2016 and July 31, 2016 as XML:

```
https://www.cuetoems.com/cems_demo/api.ashx?u=api&a=385AE3A5-C3AC-4485-9E41-81B15E2E513F&format=xml&action=volprofiles&mindate=2016-07-01&maxdate=2016-07-31
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Volunteers>
  <Volunteer>See the volprofile section above for details.</Volunteer>
</Volunteers>
```

## Code Samples

Each sample will show the **volcommittees** API call, including that language's class or object representation of a volunteer committee and how you can parse and process the results.

### C# (.NET 4.7)

#### Committee Class Definition

```
public class Committee {  
  
    public int CommitteeID { get; set; }  
    public string Name { get; set; }  
    public int MembersDesired { get; set; }  
    public int MembersAssigned { get; set; }  
    public int ChairsAssigned { get; set; }  
}
```

#### Sample Code

```
using System;  
using System.Collections.Generic;  
using System.IO;  
using System.Net;  
using System.Xml.Serialization;  
  
static void Main( string[] args ) {  
  
    // Instead of hard coding these values, consider using app.config, web.config  
    // or a database if you're using multiple sites  
    string apiAddress = "https://www.cuetoems.com/cems_demo/api.ashx";  
    string apiUsername = "api";  
    string apiAccessCode = "385AE3A5-C3AC-4485-9E41-81B15E2E513F";  
  
    // Create a list item that will hold the results and a string for the raw data  
    List<Committee> committees = null;  
    string strResponse = string.Empty;  
  
    // Make a WebRequest object for the API, passing along the username, access code,  
    // data format (XML) and the action.  
    // Use a StreamReader to read the response and save it to the string object  
    try {  
  
        WebRequest request = WebRequest.Create(apiAddress + "?u=" + apiUsername + "&a=" +  
        apiAccessCode + "&format=xml&action=volcommittees");  
        request.Timeout = -1;  
  
        using ( WebResponse response = request.GetResponse() ) {  
  
            using ( StreamReader sr = new StreamReader(response.GetResponseStream()) ) {  
                strResponse = sr.ReadToEnd();  
            }  
        }  
    }  
}
```

```

    }
}
}

catch ( Exception ex ) {

    Console.WriteLine("Error making web request: " + ex.Message);
    Console.ReadLine();

    Environment.Exit(0);
}

// Since we used XML in the request, we use the XML deserializer to process the
// raw data into an easier to use class
if ( !String.IsNullOrEmpty(strResponse) ) {

    try {

        XmlSerializer xs = new XmlSerializer(typeof(List<Committee>));

        using ( StringReader sr = new StringReader(strResponse) ) {

            committees = (List<Committee>)xs.Deserialize(sr);
        }
    }

    catch ( Exception ex ) {

        Console.WriteLine("Error deserializing data: " + ex.Message);
        Console.ReadLine();

        Environment.Exit(0);
    }
}

// Now you have the data in an easy to use class, here's an example of how
// you would iterate over the data and count the total committee assignments
int total = 0;

foreach ( Committee committee in committees ) {

    total += committee.MembersAssigned;
}
}

```