



Community Maintenance Guide

Programmatically Adding Events

For Community Software version 4.2+
Revision 1.2

July 14, 2021



CommunityWFM

phone 877-668-6870
web CommunityWFM.com

3400 Waterview Parkway, Suite 101
Richardson, Texas 75080



Table of Contents

Table of Contents 2

About this Document..... 3

Community Event Integration..... 4

 API Interface 4

 Required Authentication Header..... 5

 Name/Value Pairs..... 5

 Sample class declaration for the name/value pairs..... 8

 The RecurringEvent class definition..... 11



About this Document

The objective of this document is to describe the method of programmatically adding Scheduled Events such as PTO to the Community database via the CommunityWebAPI. Using the RESTful API functions requires that the reader understand how to create HTTP requests and how to consume JSON formatted data.



Community Event Integration

Community supports an API function exposed by the CommunityWebAPI RESTful service. The information below should enable you to successfully add Events via the RESTful service programmatically.

This interface describes creating "recurring events," which can be single-day events (vacation, sick, etc.) or events that recur over a long period of time, but only on specific days (e.g., between January 1 and March 31, I have a recurring doctor's appointment every Friday between 9:00 AM and 11:00 AM).

In cases where the event is a single day event, simply supply the same *from* and *thru* dates and times, but ensure that the d? flags are all set to true.

In the more complex example, the start date value would be "2017/01/01"; end date value would be "2017/03/31"; start time value would be "09:00:00"; end time value would be "11:00:00"; and the d? flags would be (assuming Monday start day of week) d1=false d2=false d3=false d4=false d5=true d6=false d7=false.

API Interface

The verb is POST and the entry point is

<http://<servername>/CommunityWebApi/api/SingleRecurringEvent>

Execution of the call returns a WFMSG standard WebApiResult object (in JSON format) which includes the following members:

Success – boolean indicates that the operation was successful.

Data – if successful, contains the newly created RecurringEvent (described below) object. If not successful, this member is NULL.

Message – if not successful, contains the exception thrown; otherwise this member is empty.

ExceptionResult – if an exception is thrown, this is the full body of the exception; otherwise, this member is NULL.

*Note that Community supports the use of concurrent CommunityWebAPI deployments for integration within a secure architecture as well as DMZ or dedicated appliance deployment for external interfaces.



Required Authentication Header

Community Web API release 4.3 and above require an authentication token for each entry point passed as a custom header value. Client applications can retrieve the initial authentication token from the AuthenticateEx (described in another document) POST call in a member variable named "authToken." Client applications then pass this value in the custom header "WFMSGAPIKEY" on subsequent calls. Note that the authentication token will time out after 20 minutes.

Additionally, client applications may retrieve the "refreshed" version of the token on each call by reading the custom header value of the same name ("WFMSGAPIKEY").

Name/Value Pairs

The following table contains the name/value pairs that must be included in the form submission for the RESTful call. Note that for inserting new agents into the database, the transactionFlagId value must be set to 1.

Class Member	Type	Req?	Description
transactionFlagId	Int	Yes	Indicates the type of POST transaction to perform. Valid values are: 1: Insert
agentId	Int	Yes	Sa_agent_id to assign to the newly created event(s).
eventTypeId	Int	Yes	Sa_exception_type_id of the events.
startTime	DateTime	Yes	A date/time pair in YYYY/MM/DD HH:MM (24 hour clock) format that indicates the starting time of the event. Note that, for this member, the application ignores the date portion. Example: start an event at 8:00 AM 1900-01-01 08:00:00



endTime	DateTime	Yes	<p>A date/time pair in YYYY/MM/DD HH:MM (24 hour clock) format that indicates the ending time of the event. Note that, for this member, the application ignores the date portion.</p> <p>Example: end an event at 5:00 PM 1900-01-01 17:00:00</p>
description	String	No	Provides the description field on the event
location	String	No	Provides the location field on the event
scheduleId	Int	No	If NULL, then this event is placed on the published schedule. If not NULL, then this event is placed on the working schedule where sa_schedule_id = this value.
startDate	DateTime	Yes	<p>A date value in YYYY/MM/DD format that indicates the start date of the event. Note that, for this member, the application ignores the time portion (if supplied).</p> <p>Example: start the event on May 1, 2017 2017/05/01</p>
endDate	DateTime	Yes	<p>A date value in YYYY/MM/DD format that indicates the end date of the event. Note that, for this member, the application ignores the time portion (if supplied).</p> <p>Example: end the event on May 4, 2017 2017/05/04</p>



d1	Bool	Yes	A boolean value that indicates if this event should occur on the first day of the week (typically Monday).
d2	Bool	Yes	A boolean value that indicates if this event should occur on the second day of the week.
d3	Bool	Yes	A boolean value that indicates if this event should occur on the third day of the week.
d4	Bool	Yes	A boolean value that indicates if this event should occur on the fourth day of the week.
d5	Bool	Yes	A boolean value that indicates if this event should occur on the fifth day of the week.
d6	Bool	Yes	A boolean value that indicates if this event should occur on the sixth day of the week.
d7	Bool	Yes	A boolean value that indicates if this event should occur on the seventh day of the week.
timeZoneId	Int	No	The time zone of the event; typically, this will be the agent's local time but if NULL then the application defaults to corporate time zone.
responsibleUserId	Int	Yes	The sa_agent_id of the user with sufficient privilege to make changes to this agent's schedule.



Sample class declaration for the name/value pairs

(as described in the table above)

```
public class SingleRecurringEventPostParams
{
    /// <summary>
    /// transactionFlagId dictates how the event will be modified.
    /// </summary>
    public int transactionFlagId { get; set; }

    /// <summary>
    /// agentId
    /// </summary>
    public int agentId { get; set; }

    /// <summary>
    /// eventTypeId
    /// </summary>
    public int eventTypeId { get; set; }

    /// <summary>
    /// startTime
    /// </summary>
    public DateTime startTime { get; set; }

    /// <summary>
    /// endTime
    /// </summary>
    public DateTime endTime { get; set; }

    /// <summary>
    /// description
    /// </summary>
    public string description { get; set; }

    /// <summary>
    /// location
```



```
/// </summary>
public string location { get; set; }

/// <summary>
/// scheduleId
/// </summary>
public int? scheduleId { get; set; }

/// <summary>
/// startDate
/// </summary>
public DateTime startDate { get; set; }

/// <summary>
/// endDate
/// </summary>
public DateTime endDate { get; set; }

/// <summary>
/// d1
/// </summary>
public bool d1 { get; set; }

/// <summary>
/// d2
/// </summary>
public bool d2 { get; set; }

/// <summary>
/// d3
/// </summary>
public bool d3 { get; set; }

/// <summary>
/// d4
/// </summary>
public bool d4 { get; set; }
```



```
/// <summary>
/// d5
/// </summary>
public bool d5 { get; set; }

/// <summary>
/// d6
/// </summary>
public bool d6 { get; set; }

/// <summary>
/// d7
/// </summary>
public bool d7 { get; set; }

/// <summary>
/// timeZoneId
/// </summary>
public int? timeZoneId { get; set; }

/// <summary>
/// Responsible party user id - used for audit purposes.
/// </summary>
public int? responsibleUserId { get; set; }
}
```



The RecurringEvent class definition

(WebApiResult.Data member of POST verb with the above parameters)

```
#region Class Members

[DataMember]
/// <summary>
/// From Get Procedure, parameter @StartTime [6] [Not a Key]
/// </summary>
public DateTime? StartTime { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @CreateDate [18] [Not a Key]
/// </summary>
public DateTime? CreateDate { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @AssignmentType [20] [Not a Key]
/// </summary>
public RecurringEventAssignmentTypes AssignmentType { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @RecurringEventId [0] [Not a Key]
/// </summary>
public int? RecurringEventId { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @EndDate [8] [Not a Key]
```



```
/// </summary>
public DateTime? EndDate { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @AllDay [9] [Not a Key]
/// </summary>
public bool? AllDay { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @TimeZoneld [0] [Not a Key]
/// </summary>
public int? TimeZoneld { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @ScheduleId [1] [Not a Key]
/// </summary>
public int? ScheduleId { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @ProcessAgentId [23] [Not a Key]
/// </summary>
public int? ProcessAgentId { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @ProcessDate [22] [Not a Key]
/// </summary>
public DateTime? ProcessDate { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @Description [5] [Not a Key]
```



```
/// </summary>
public string Description { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @StartDate [7] [Not a Key]
/// </summary>
public DateTime? StartDate { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @ExceptionTypeId [2] [Not a Key]
/// </summary>
public int? ExceptionTypeId { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @BaseDate [17] [Not a Key]
/// </summary>
public DateTime? BaseDate { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek4 [13] [Not a Key]
/// </summary>
public bool? IncludeDayOfWeek4 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek5 [14] [Not a Key]
/// </summary>
public bool? IncludeDayOfWeek5 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek6 [15] [Not a Key]
```



```
/// </summary>
public bool? IncludeDayOfWeek6 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek7 [16] [Not a Key]
/// </summary>
public bool? IncludeDayOfWeek7 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @Location [4] [Not a Key]
/// </summary>
public string Location { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek1 [10] [Not a Key]
/// </summary>
public bool? IncludeDayOfWeek1 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek2 [11] [Not a Key]
/// </summary>
public bool? IncludeDayOfWeek2 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @IncludeDayOfWeek3 [12] [Not a Key]
/// </summary>
public bool? IncludeDayOfWeek3 { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @LastUpdate [19] [Not a Key]
```



```
/// </summary>
public DateTime? LastUpdate { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @Duration [3] [Not a Key]
/// </summary>
public double? Duration { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @ProcessAgentName [24] [Not a Key]
/// </summary>
public string ProcessAgentName { get; set; }

[DataMember]
/// <summary>
/// From Get Procedure, parameter @Processed [21] [Not a Key]
/// </summary>
public bool? Processed { get; set; }

public DataTable ValidationResults { get; set; }

public DataTable AdjustedEvents { get; set; }

public DataTable CreatedEvents { get; set; }

/// <summary>
/// Create a dynamic key for use in audit services.
/// </summary>
private dynamic key = new System.Dynamic.ExpandoObject();

#endregion
```