



Community Maintenance Guide

Programmatically adding agents

For Community Software version 4.2+

Revision 1.2

July 19, 2021



phone 877-668-6870

web CommunityWFM.com

3400 Waterview Parkway, Suite 101

Richardson, Texas 75080

Table of Contents

| | |
|---|---|
| About this Document | 3 |
| Community Agent Integration | 4 |
| Community Stored Procedure Method | 4 |
| Foreign Keys | 4 |
| Stored Procedure | 5 |
| API Method | 6 |
| API Interface | 6 |
| Name/Value Pairs..... | 7 |

About this Document

The objective of this document is to describe the methods of programmatically adding agents to the CommunityWFM database. CommunityWFM supports both a stored procedure method and an API. To use the database interface, this document assumes the reader has some familiarity with T-SQL and SQL Server clients. Additionally, using the RESTful API functions requires an understanding of how to create HTTP requests and how to consume JSON formatted data.

Community Agent Integration

Community supports two methods of programmatically adding agents to the database: A stored procedure method, and an API function exposed by the CommunityWeb API RESTful service.

Community Stored Procedure Method

With access to the Community3 database users may access the stored procedure `spAgentsInsert`, to add user rows to the `t_agent_table`. The signature of the procedure is included below. Implemented in this procedure are foreign-key constraints that you will need to be aware of. Each foreign-key is described in the table below.

Foreign Keys

| | |
|----------------------------|---|
| Supervisor_id | FK to <code>t_agent</code> table identifying this agent's supervisor, or NULL if none |
| Sa_activity_id | FK to <code>t_activity</code> identifying the site that the agent is assigned to. Cannot be NULL |
| Work_type | FK to <code>t_exception_types</code> table identifying the default work event type for this agent. Cannot be NULL |
| Sa_role_id | FK to <code>t_roles</code> table identifying the system role this agent has. 100=Agent, 200=Supervisor, 300=Scheduler and 400=Administrator. Cannot be NULL |
| Sa_timezone_id | FK to <code>t_timezones</code> table allowing a custom time zone. If NULL, this is inherited from the site definition. |
| EmpClassificationId | FK to <code>t_employee_classifications</code> . Can be NULL |
| ResponsibleAgentId | Identifies the user responsible for making this change. Used for audit purposes and must be an entry in <code>t_agent</code> |

Stored Procedure

The spAgentsInsert procedure has the following signature:

```
ALTER PROCEDURE [dbo].[spAgentsInsert]

    (@first_name    name

    ,@last_name     name

    ,@middle_initial name

    ,@title         name

    ,@email_address    description

    ,@employee_id    name

    ,@supervisor_id  int

    ,@sa_activity_id int

    ,@hire_date      datetime = NULL

    ,@work_type      int

    ,@login_id       name

    ,@tiebreaker     int

    ,@sa_role_id     int

    ,@sa_timezone_id int

    ,@EmpClassificationId int

    ,@ResponsibleAgentId int = NULL

    ,@ExternalUserId nvarchar(255) = NULL

    ,@UserTags nvarchar(max)

    ,@sa_agent_id int OUTPUT)
```

API Method

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

API Interface

The verb is POST and the entry point is http://<servername>/CommunityWebApi/api/agent*

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

Success – boolean indicates that the operation was successful.

Data – if successful, contains the newly created Agent 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:** Community supports the use of concurrent CommunityWeb API deployments for integration within a secure architecture as well as DMZ or dedicated appliance deployment for external interfaces.

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 | <p>Indicates the type of POST transaction to perform. Valid values are:</p> <ul style="list-style-type: none"> 1: Insert a new Agent (t_agent) row 2: Update an existing Agent row 3: Delete an existing Agent row 4: Update default work type for an existing Agent row 5: Update login ID 6: Update password 7: Import from 3rd party system (eg. Staging Database) |
| id | Int? | No | <p>Provides the unique sa_agent_id (from t_agent)</p> <ul style="list-style-type: none"> • Ignored for INSERT • Required for all other POST transactions |
| defaultWorkTypeId | Int? | No | <p>Provides the default work type for schedules for this agent (user). This is a foreign key value to the t_exception_types table, and is NOT NULL on the t_agent table.</p> <ul style="list-style-type: none"> • Required for INSERT • Required for UPDATE • Required for UPDATE default work type • Ignored for all other POST transactions |

| | | | |
|---------------------------------|--------|-----|---|
| supervisorId | Int? | No | Provides the agent's supervisor id. This is a NULLable column in the t_agent table. Additionally, it is a same-table foreign key to the t_agent table. <ul style="list-style-type: none"> Optional for all POST transactions. If NULL, then this agent is considered to be at the "top" of the organizational chart. |
| timeZoneId | Int | No | Indicates the agent's time zone. If NULL, then the default time zone will be used (as defined by the agent's site). |
| employeeClassificationId | Int | No | Indicates the employee classification assigned to this agent. If NULL, then no classification will be assigned. |
| siteId | Int | No | Indicates the site to which this agent belongs. Foreign key to t_activity table. |
| firstName | String | Yes | First name |
| lastName | String | Yes | Last name |
| middleInitial | String | Yes | Middle initial |
| hireDate | String | Yes | Hire date; must be in format YYYY/MM/DD |
| employeeId | String | Yes | Employee id |
| title | String | Yes | Title |
| emailAddress | String | Yes | Email address |
| roleId | Int | No | Agent's role within the application. Foreign key to t_roles table. |
| tiebreakValue | Int | No | Tiebreak value |
| loginId | String | Yes | Application login ID (note this is not the ACD login ID). |
| Password | String | Yes | Password for login |

| | | | |
|------------------------|--------|-----|--|
| externalLoginId | String | Yes | For environments that use Active Domain / Integrated Authentication, this is the login ID used to connect external clients to the application. Mainly used for the Community Everywhere mobile applications. |
| userTags | String | Yes | Collection of hash-tags associated with the agent. |
| userID | Int | No | User ID responsible for making the changes to this agent. Used for security and audit purposes. |

Sample class declaration for the name/value pairs described in the table above

```
public class AgentPostParams
{
    /// <summary>
    /// transactionFlagId decides how to modify agent.
    /// </summary>
    public int transactionFlagId { get; set; }

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

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

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

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

```
/// timeZoneId
```

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

```
public int? timeZoneId { get; set; }
```

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

```
/// employeeClassificationId
```

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

```
public int? employeeClassificationId { get; set; }
```

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

```
/// siteId
```

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

```
public int? siteId { get; set; }
```

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

```
/// firstName
```

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

```
public string firstName { get; set; }
```

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

phone 877-668-6870

web CommunityWFM.com

3400 Waterview Parkway, Suite 101

Richardson, Texas 75080

```
/// lastName
```

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

```
public string lastName { get; set; }
```

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

```
/// middleInitial
```

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

```
public string middleInitial { get; set; }
```

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

```
/// hireDate
```

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

```
public string hireDate { get; set; }
```

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

```
/// employeeld
```

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

```
public string employeeld { get; set; }
```

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

```
/// title
```

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

```
public string title { get; set; }
```

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

```
/// emailAddress
```

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

```
public string emailAddress { get; set; }
```

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

```
/// roleId
```

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

```
public int? roleId { get; set; }
```

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

```
/// tiebreakValue
```

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

```
public int? tiebreakValue { get; set; }
```

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

```
/// loginId
```

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

```
public string loginId { get; set; }
```

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

```
/// password
```

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

```
public string password { get; set; }
```

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

```
/// externalLoginId
```

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

```
public string externalLoginId { get; set; }
```

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

```
/// User tags
```

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

```
public string userTags { get; set; }
```

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

```
/// For security purposes.
```

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

```
public int? userId { get; set; }
```

```
}
```