

# SPARK



## Workflow for SharePoint API Technical Manual

Ver. 6.9.35.105

**SPARK**   
workflow

Improve the way you work ....

**SPARK**nit Inc.  
11420 Delores Ferguson Ln  
Charlotte, NC 28277  
Tel: +1 (704) 559-5950

 [info@sparknit.com](mailto:info@sparknit.com) |  [www.sparknit.com](http://www.sparknit.com)

## Introduction

---

This technical manual provides detailed information on how to use SPARK Workflow API to manage and control SPARK Tasks and functionalities from a third party application or solution.

## Before you begin

---

Before you begin, you need to consider the followings:

- Please note that you need to import SPARKWorkflow.dll to your solution assemblies; you will find it in your SharePoint Application servers in the GAC folder, i.e. C:\Windows\Microsoft.NET\assembly\GAC\_MSIL\SPARKWorkflow.
- This API works only on any SharePoint solution, Event Receiver project, web part or SharePoint supported classes.
- You should use SPARKWorkflow.dll file that is designate for the specific SharePoint edition.

## SPARKWorkflowTask Class

---

Namespace: SPARKWorkflow.API

Assembly: SPARKWorkflow.dll

[[SPARKWorkflow.API.SPARKWorkflowTask](#)]

C#

```
public SPARKWorkflowTask(string TaskListID, int TaskItemID);  
public SPARKWorkflowTask(string TaskListID, string TaskItemGUID);
```

### Constructors:

SPARKWorkflowTask()

### Properties:

Property Name	TaksListID
Property Value	String
Description	The string value of the List's GUID

Property Name	TaksItemID
Property Value	Integer
Description	The ID of the task

Property Name	TaksItemGUID
Property Value	String
Description	The string representing the GUID of the task

### Example:

C#

```
SPARKWorkflowTask Task = new SPARKWorkflowTask("0c0f282c-d0cc-45e4-8d53-  
b0e5a9931e06", 1607);
```

C#

```
SPARKWorkflowTask Task = new SPARKWorkflowTask("0c0f282c-d0cc-45e4-8d53-  
b0e5a9931e06", "0c0f282c-d0cc-45e4-8d53-b0e5er55");
```

## GetTaskAssignedTo Function

---

Namespace: SPARKWorkflow.API

Assembly: SPARKWorkflow.dll

[[SPARKWorkflow.API.SPARKWorkflowTask.GetTaskAssignedTo](#)]

Description: Returns the assigned to user of the designated task.

```
C#  
public Microsoft.SharePoint.SPUser GetAssignedTo();
```

### Constructors:

GetTaskAssignedTo()

### Properties:

None

### Example:

```
C#  
SPUser AssignTo= Task.GetTaskAssignTo()
```

## RetrieveCustomOutcomes Function

---

Namespace: SPARKWorkflow.API

Assembly: SPARKWorkflow.dll

[[SPARKWorkflow.API.SPARKWorkflowTask.RetrieveCustomOutcomes](#)]

Description: Returns the custom outcomes of the designated task in an array.

```
C#  
public System.Collections.ArrayList RetrieveCustomOutcomes();
```

### Constructors:

RetrieveCustomOutcomes()

### Properties:

None

### Example:

```
C#  
ArrayList OutcomesList= Task.RetrieveCustomOutcomes()
```

## GetSPARKTaskType Function

---

Namespace: SPARKWorkflow.API

Assembly: SPARKWorkflow.dll

[[SPARKWorkflow.API.SPARKWorkflowTask.GetSPARKTaskType](#)]

Description: Returns SPARK task type [SPARK Approval | SPARK Review | SPARK Custom] as string for the designated task.

```
C#  
public String GetSPARKTaskType();
```

### Constructors:

GetSPARKTaskType()

### Properties:

None

### Example:

```
C#  
string TaskType= Task.GetSPARKTaskType()
```

## DelegateSPARKTask Function

Namespace: SPARKWorkflow.API

Assembly: SPARKWorkflow.dll

[[SPARKWorkflow.API.SPARKWorkflowTask.DelegateSPARKTask](#)]

Description: Delegates the designated task to another user.

```
C#  
public void DelegateSPARKTask(SPUser DelegateBy, SPUser DelegateTo, string  
DelegateComment)
```

### Constructors:

DelegateSPARKTask()

### Properties:

Property Name	DelegateBy
Property Value	SPUser
Description	Represents the user that initiated the delegation process.

Property Name	DelegateTo
Property Value	SPUser
Description	Represents the user to delegate the designated task to.

Property Name	DeleagteComment
Property Value	String
Description	Task delegate comment.

### Example:

```
C#  
SPUser DelegateTo=web.EnsurUser("Login Name")  
Task.DelegateSPARKTask(web.CurrentUser, DelegateTo, "Delegate Comment");
```

## RespondToSPARKTask Function

Namespace: SPARKWorkflow.API

Assembly: SPARKWorkflow.dll

[[SPARKWorkflow.API.SPARKWorkflowTask.RespondToSPARKTask](#)]

Description: Respond to the designated SPARK task regardless of its type. The task will be completed after executing this function.

```
C#
public void RespondToSPARKTask(string TaskComments, string TaskStatus, string
TaskOutcome, SPARKTaskType TypeOfTask, Boolean OnlyAssignedToUserCanRespond)
```

### Constructors:

RespondToSPARKTask()

### Properties:

Property Name	TaskComments
Property Value	string
Description	Task respond comments

Property Name	TaskStatus
Property Value	string
Description	Represents the task status to be updated after executing the function, usually it will be "Completed" but you can have it customized or resourced based on User Language Culture.

Property Name	TaskOutcome
Property Value	string
Description	Represents the responded outcome of the designated task.

Property Name	TypeofTask
Property Value	SPARKTaskType
Description	Represents the task type's enumerated values [Approve, Reject, Review and Custom].

Property Name	OnlyAssignedToUserCanRespond
Property Value	Boolean
Description	If the Boolean value is "true" then it will restrict responding to this task to only "Assigned To" user, no other user will be able to respond to this task, an exception will be thrown in the workflow history log. If the Boolean value is "false", any user will be able to respond to the task.



## Examples:

```
C#
```

```
Test.RespondToSPARKTask("Your Comment", "Completed", "Outcome3",  
SPARKTaskType.Custom, true);
```

```
C#
```

```
Test.RespondToSPARKTask("Your Comment", "Completed", "test3", SPARKTaskType.Approve,  
true);
```

```
C#
```

```
void RunApprovalTask()  
{  
    try  
    {  
        SPARKWorkflowTask Task = new SPARKWorkflowTask("0c0f282c-d0cc-45e4-  
8d53-b0e5a9931e06", 1611);  
        using (SPSite site = new SPSite(SPContext.Current.Web.Url))  
        using (SPWeb web = site.OpenWeb())  
        {  
            Task.DelegateSPARKTask(web.CurrentUser, web.CurrentUser,  
"Delegate To George");  
            string TaskType = Task.GetSPARKTaskType();  
            if (TaskType == "SPARK Approval") {  
                Task.RespondToSPARKTask("Approval Comment", "Completed",  
"Approve", SPARKTaskType.Approve, true);  
            }  
            else if (TaskType == "SPARK Review")  
            {  
                Task.RespondToSPARKTask("Review Comment", "Completed",  
"Continue", SPARKTaskType.Review, true);  
            }  
            else if (TaskType == "SPARK Custom")  
            {  
                ArrayList OutcomesList = Task.RetrieveCustomOutcomes("", 0);  
                foreach (string outcome in OutcomesList)  
                {  
                    if(outcome=="outcome1")  
                        Task.RespondToSPARKTask("Custom Comment",  
"Completed", outcome, SPARKTaskType.Custom, true);  
                }  
            }  
        }  
    }  
    catch (Exception ex)  
    {  
        Response.Write(ex.Message);  
    }  
}
```