



Select Survey .NET

API Documentation



Table of Contents

SelectSurvey.NET API Developer Reference.....	2
Example "Survey API Console" Application	2
Current SelectSurvey.NET API features:	2



SelectSurvey.NET API Developer Reference

SelectSurvey.NET includes an API (Application Programming Interface) web service which allows 3rd party programs to communicate with the SelectSurvey.NET application. This document contains developer references and feature list and also a sample application is provided with SelectSurvey.NET which demonstrates to developers how to access the API.

The API is protected by credentials that are encrypted in each method call. So to call any method, you have to use a valid user name/password. These users must be valid and existing **SelectSurvey.NET** users to access the API. Full source code is also available for purchasing for those companies that require extensive customizations.

To access the API, you need to go to the service URL and pass the credentials for login.

Example: <http://10.selectsurvey.net/Demo-EN/Services/SurveyAPI.asmx>

Example "Survey API Console" Application

An example application which communicates with the SelectSurvey.NET API is included in the SelectSurvey.NET zip file that you received when you purchased the product. In the zip file there is a folder "SurveyAPIConsole" which contains the sample application which demonstrates usage of the API. Programmers can use this as a guide on how to set up their own programs or 3rd party application to communicate with the SelectSurvey.NET application.

This sample application is written in C#.net for .net 4.5 framework. All source code is available for you to view to learn how the API works. The source code is located in the folder "SurveyAPIConsole" and can be opened with Visual Studio 2015.

Current SelectSurvey.NET API features:

1. AddRecipientToEmailList

```
/// Add Recipient to Email List
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="emailListId">Id of the email list to delete</param>
/// <param name="emailAddress">Email Address to add</param>
/// <param name="fName">First Name of the email holder</param>
/// <param name="lName">Last Name of the email holder</param>
```

```
/// <param name="custom1">Custom Data1 for that email</param>
/// <param name="custom2">Custom Data2 for that email</param>
/// <param name="custom3">Custom Data3 for that email</param>
```

2. [AddSurveyToSurveyFolder](#)

```
/// AddSurveyToSurveyFolder
/// </summary>
/// <param name="credentials"></param>
```

3. [CopyQuestionItemToSurvey](#)

```
/// copies item from one survey to another by the item id.
/// </summary>
/// <param name="credentials"></param>
/// <param name="itemId"></param>
/// <param name="surveyId"></param>
/// <param name="orderNumber"></param>
/// <param name="pageNumber"></param>
```

4. [CopyResponseAnswer](#)

```
/// copies response answer fields from one response id to another response id.
/// </summary>
/// <param name="credentials"></param>
/// <param name="itemId"></param>
/// <param name="surveyId"></param>
/// <param name="orderNumber"></param>
/// <param name="pageNumber"></param>
```

5. [CopySurvey](#)

```
/// Creates a copy of a survey.
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="originalSurveyID">ID of original survey to be copied from</param>
/// <param name="newSurveyName">Title for new survey</param>
/// <param name="ownerUsername">The sur_user.username value of the new survey owner or string
of usernames separated by semicolons.</param>
/// <returns>The id of the newly created survey</returns>
```

6. [CreateEmailList](#)

```
/// Creates a new email list.
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="listName">Name of the list</param>
/// <param name="description">Description of the list</param>
/// <param name="isDefault">true if the list is the default list</param>
```

```
/// <returns>The id of the newly created email list</returns>
```

7. CreateResponseID

```
/// Creates a response ID with default information for todays date and increments the response  
count in the survey table.  
/// </summary>  
/// <param name="credentials"></param>  
/// <param name="surveyID"></param>  
  
/// <returns>Response ID</returns>
```

8. CreateSurveyFolder

```
/// CreateSurveyFolder  
/// </summary>  
  
/// <param name="credentials"></param>
```

9. DeleteEmailList

```
/// Deletes the email list specified by the id  
/// </summary>  
/// <param name="credentials">User Credentials</param>  
  
/// <param name="emailListId">Id of the email list to delete</param>
```

10. DeleteResponseID

```
/// Deletes a response ID from the sur_response table. Does not delete the answers associated with  
this response.  
/// </summary>  
/// <param name="credentials"></param>  
/// <param name="responseID"></param>  
  
/// <returns>Bool true if responseID was deleted.</returns>
```

11. DeleteResponseIDandAssociatedAnswers

```
/// Deletes the specified response ID and all answers associated with that response ID.  
/// </summary>  
/// <param name="credentials"></param>  
/// <param name="responseID"></param>  
/// <param name="surveyID"></param>  
  
/// <returns>Bool true if response ID and answers were deleted.</returns>
```

12. DeleteSurveyAndAllAssociatedData

```
/// Deletes a survey and all associated data including responses, page conditions, pages, question  
items, etc.  
/// </summary>  
/// <param name="credentials"></param>
```

```
/// <param name="surveyId"></param>
```

13. [DeleteSurveyFolder](#)

```
/// DeleteSurveyFolder  
/// </summary>  
/// <param name="credentials"></param>
```

14. [GetEmailListRecipients](#)

```
/// Gets the list of recipients for the specified email list  
/// </summary>  
/// <param name="credential">User Credenails</param>  
/// <param name="emailListId">Id of the list to retrieve</param>  
/// <returns>Array of EmailListRecipient objects</returns>
```

15. [GetEmailLists](#)

```
/// Gets a list of email lists.  
/// </summary>  
/// <param name="credentials">User Credentials</param>  
/// <returns>Array of EmailList objects</returns>
```

16. [GetQuestionItemListByLibraryID](#)

```
/// Returns an array of question item information: itemid,itemtypeid,itemtext  
/// </summary>  
/// <param name="credentials"></param>  
/// <param name="libraryId"></param>
```

17. [GetQuestionItemListBySurveyID](#)

```
````/// Returns an array of question item information: itemid,itemtypeid,itemtext,itemalias  
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
```

### 18. [GetResponseListByResponseID](#)

```
/// Returns an array of response answer information:
response_id,item_id,subitem_id,answer_id,answer_text,other_text,column_answer_id,date_updated,evaluation
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
```

### 19. [GetResponseListBySurveyID](#)

```
/// Returns an array of response answer information:
response_id,item_id,subitem_id,answer_id,answer_text,other_text,column_answer_id,date_updated,eval
ution
/// </summary>
/// <param name="credentials"></param>

/// <param name="surveyId"></param>
```

## 20. [GetResponseListBySurveyIDAndModifiedDate](#)

```
/// Returns an array of response answer information:
response_id,item_id,subitem_id,answer_id,answer_text,other_text,column_answer_id,date_updated,eval
ution
/// </summary>
/// <param name="credentials"></param>

/// <param name="surveyId"></param>
```

## 21. [GetSurveyDeploymentOptions](#)

```
/// Retrieves the deployment options for the specified survey.
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="surveyId">Id of the survey to generate the deployment options for</param>

/// <returns>String array of the three deployment options. Index = 0, email link. Index = 1, web
link. Index = 3, pop-up link.</returns>
```

## 22. [GetSurveyFolderList](#)

```
/// Retrieves a list of Survey folders
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <returns>List of TextValuePair objects</returns>
```

## 23. [GetSurveyIDsByEmailAddress](#)

```
/// Gets ArrayList of survey ID and Survey Title
/// </summary>
/// <param name="credentials"></param>
/// <param name="email"></param>
/// <returns>ArraryList each row contains: surveyid|surveytitle (separated by a pipe
symbol)</returns>
```

## 24. [GetSurveyList](#)

```
/// Retrieves a list of surveys in the specified folder.
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="surveyFolderId">Id of the folder to get the surveys from</param>
/// <param name="sortColumn">Enum of the sort columns</param>
/// <param name="sortDirection">Enum of the sort direction</param>

/// <returns>Array of SurveyInfo. Result may be null.</returns>
```

## 25. [GetUnapprovedSurveys](#)

```
/// GetUnapprovedSurveys
/// </summary>
/// <param name="credentials"></param>

/// <returns> ArrayList of unapproved survey_id </returns>
```

## 26. [ImportEmailListRecipients](#)

```
/// Imports a list of EmailListRecipient into the specified email list
/// </summary>
/// <param name="credential">User Credentials</param>
/// <param name="emailListId">Id of the email list to import into</param>
/// <param name="recipients">Array of EmailListRecipient</param>

/// <returns>Number of recipients imported</returns>
```

## 27. [InsertPageCondition](#)

```
/// Insert Page Condition
/// </summary>
/// <param name="credentials"></param>
/// <param name="page_condition_id"></param>
/// <param name="survey_id"></param>
/// <param name="page_number"></param>
/// <param name="dependent_item_id"></param>
/// <param name="group_number">1</param>
/// <param name="operator_id">1</param>
/// <param name="answer_id">-1</param>
/// <param name="answer_text">Yes</param>
/// <param name="skip_to_page">-1</param>
/// <param name="skip_to_end">N</param>
/// <param name="dependent_subitem_id">0</param>
/// <param name="token">-1</param>

/// <param name="column_answer_id">0</param>
```

## 28. [RemoveRecipeintFromEmailList](#)

```
/// Remove recipeint from Email List
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="emailListId">Id of the email list to delete</param>

/// <param name="emailAddressId">Id of the email address to delete</param>
```

## 29. [SendFollowUpEmail](#)

```
/// Send Follow Up Email Message (This method mimics the actions on the SendFollowUpMessage2.aspx
page). Works with or without ResponseLogic.
/// </summary>
/// <param name="credentials"></param>
```



```
 /// <param name="strNoResponse">pass in string "on" to include Non Respondents, otherwise empty
string.</param>
 /// <param name="strDeclined">pass in string "on" to include recipients that clicked decline link,
otherwise empty string.</param>
 /// <param name="strResponded">pass in string "on" to include recipients that started the survey,
otherwise empty string.</param>
 /// <param name="strCompleted">pass in string "on" to include recipients that completed, otherwise
empty string.</param>
 /// <param name="strExcludeOptOut">pass in string "on" to exclude recipients that clicked to opt
out, otherwise empty string.</param>
 /// <param name="strExcludeInactive">pass in string "on" to exclude recipients that have been
marked inactive in the email list since the original send (typically from a bounce or invalid
email), otherwise empty string.</param>
 /// <param name="strEmailMessageId">Numeric ID of the original email message this is a follow up
for.</param>
 /// <param name="subject">Email subject</param>
 /// <param name="strTextBody">both plain text and html version of the email message are
required for multi part mime</param>
 /// <param name="strHtmlBody">both plain text and html version of the email message are required
for multi part mime</param>
 /// <param name="strFromName">From Name</param>
 /// <param name="strFromEmail">From Email Address</param>
 /// <param name="strCCAddress">CC Address</param>
 /// <param name="strBCCAddress">BCC Address</param>
 /// <param name="strSendDate">Send Date.If empty will send now.</param>
 /// <param name="strSendTime">Send Time</param>
```

### 30. SendSurveyByEmailList

```
/// Send the specified survey to the recipients in the specified email list. Works with or without
ResponseLogic.
/// </summary>
/// <param name="credentials">User Credentials</param>
/// <param name="surveyId">Id of the survey to send</param>
/// <param name="emailListId">Id of the email list to send the survey to</param>
/// <param name="subject">Subject of the email</param>
/// <param name="textBody">Plain text of the email</param>
/// <param name="htmlBody">HTML text of the email</param>
/// <param name="fromName">From name</param>
/// <param name="fromEmail">From email address</param>
```

### 31. UpdatePageCondition

```
/// Update Page Condition
/// </summary>
/// <param name="credentials"></param>
/// <param name="page_condition_id"></param>
/// <param name="survey_id"></param>
/// <param name="page_number"></param>
/// <param name="dependent_item_id"></param>
/// <param name="group_number">1</param>
/// <param name="operator_id">1</param>
/// <param name="answer_id">-1</param>
```

```
/// <param name="answer_text">Yes</param>
/// <param name="skip_to_page">-1</param>
/// <param name="skip_to_end">N</param>
/// <param name="dependent_subitem_id">0</param>
/// <param name="token">-1</param>

/// <param name="column_answer_id">0</param>
```

### 32. UpdateSurveyFolder

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

/// <param name="credentials"></param>
```

### 33. deleteDesignApproval

```
/// Delete the design for the given surveyId
/// </summary>
/// <param name="credentials"></param>

/// <param name="surveyId"></param>
```

### 34. deleteSurveyPageAndContents

```
/// Deletes a survey page and its contents
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>

/// <param name="pageNumber"></param>
```

### 35. getDesignApproval

```
/// gets arraylist with design approvals
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
/// <returns>ArrayList each row is a pipe delimited list in this order:
surveyId|comments|reviewer_user_id|accepted_date|accepted_yn</returns>
```

### 36. insertUpdateDesignApproval

```
/// if design approval is not in database method will do insertion, if exists the it will do
update
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
/// <param name="approveComment"></param>

/// <param name="radioApproveButtonYN"></param>
```

### 37. updateQuestionTextAndSubText

```
/// Update Question Text and Sub Text
/// </summary>
/// <param name="credentials"></param>
/// <param name="itemID"></param>
/// <param name="itemText"></param>
/// <param name="itemSubText"></param>
```

### 38. updateSurveyOwner

```
/// Update survey owner
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
/// <param name="adminEmailAddress"></param>
/// <param name="owners"></param>
/// <param name="userGroupOwners"></param>
/// <param name="adUserGroupOwners"></param>
```

### 39. updateSurveyStartAndEndTime

```
/// Update survey start and end date and time
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
/// <param name="startDateTime"></param>
/// <param name="endDateTime"></param>
```

### 40. updateSurveyTitle

```
/// Update the survey title
/// </summary>
/// <param name="credentials"></param>
/// <param name="surveyId"></param>
/// <param name="titleText"></param>
```