inwise

Web Services API

March 2008

Groups

1. *Create a new group*

   Operation name: Groups_CreateGroup
   In Parameters : *int* - the folderid in which the group should appear
   Out Parameters : *int* - the new group id

2. *Get a list of recipients for a group*

   Operation name: Groups_GetRecipients
   In Parameters : *int* - `groupid`
   Out Parameters : *DataSet*[datasets 14]

3. *Get a list of all test groups*

   Operation name: Groups_MyTestGroups
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 15]

4. *Get a list of all regular groups*

   Operation name: Groups_MyRegularGroups
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 15]

External Groups

1. *Get All External Groups Details*

   Operation name: Groups_External_GetRemoteGroups
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 1]

2. *Add New External Database Connection*

   Operation name: Groups_External_AddNewDBConnection
   In parameters: InwiseApplicationDataProvider.*ExternalConnection*[classes 1] class (Mandatory fields: connectionString; connectionName; tableName; dbType).
   Out parameters: *int* – the new Database connection id

3. *Get All External Connections Details*

   Operation Name: Groups_External_GetRemoteConnectionsDetails
   In Parameters : *None*

Out Parameters : *DataSet*[datasets 2]

4. *Create a New External Group*

Operation Name: Groups External_CreateGroup
In Parameters: *int* dbid – The Database connection id to use,
string groupname – the new group name,
string remoteTableName – the name of the table in the remote database from which to select the data,
DataSet fieldsMapping[datasets 3]
Out Parameters: *int* – the new group id

Queue

1. *Get a view of the current import queue*

Operation name: Queue_Importing
In Parameters : *None*
Out Parameters : *DataSet*[datasets 17]

2. *Get a view of the current sending queue*

Operation name: `Queue_Sending`
In Parameters : *None*
Out Parameters : *DataSet*[datasets 16]

Messages

1. *Get a Message Details*

Operation Name: Messages_GetMessage
In Parameters : *int* – the send id of the message to retreive
Out Parameters : InwiseApplicationDataProvider. *InwiseMessage*[classes 2] class

2. *Get all messages details, no message body*

Operation Name: Messages_GetAllMessagesSummary
In Parameters : *None*
Out Parameters : *array* of InwiseApplicationDataProvider.*InwiseMessage*

3. *Create a new message based on an existing one*

Operation Name: Messages_CopyMessage
In Parameters : *int* – the send id of the message to copy
Out Parameters : *int* – the newly created message sendid

4. *Create a new message*

Operation Name: Messages_CreateNewMessage
In Parameters : InwiseApplicationDataProvider.*InwiseMessage* – InwiseMessage

with all relevant details except for the sendid
Out Parameters : *int* – the newly created message sendid

5. *Get a link text by link id*

Operation Name: Messages_GetLinkText
In Parameters : *int* – the link id
Out Parameters : *int* – the link text

6. *Get the names and ids of all the groups for which the message was sent to*

Operation Name: Messages_GetSentGroups
In Parameters : *int* – `sendid`
Out Parameters : *DataSet*[datasets 15]

Recipients

1. *Check if an email is marked as unsubscribed*

Operation Name: Recipients_checkUnsubscribed
In Parameters : *string* – the recipient email
Out Parameters : *boolean* - indicates if the email is marked as unsubscribed

2. *Sign up a new recipient throught registration form*

Operation Name : Recipients_SignUpRecipient
In Parameters *: string* – the recipient email
boolean – use double opt in
DataSet[datasets 5] – recipient fields values
array[] – groups id to subscribe
boolean – send welcome mail to the new recipient
string – url of the mail body to send to the new recipient
string – url of the text to show on browser after new recipient registration
string – url of the mail body to send to the an existing recipient
string – url of the text to show on browser after existing recipient registration
string – charset
string – email subject for an existing recipient
string – email subject to a new recipient
boolean – update details for an existing recipient
boolean – send welcome mail to an existing recipient
Out Parameters : *string*- the result text to display to the recipient

3. Quick import recipients

Operation Name : Recipients_QuickImport
In Parameters *:DataSet*[datasets 7] – `recipients`
DataSet[datasets 8] – `mapping`
int – the id of the group to import to
Boolean – update details for existing
Out Parameters : *int* – the new import id in the queue

4. *Get All Messages Sent for Recipient*

   Operation Name: Recipients_GetMessages
   In Parameters : *int* recipientId – the recipient id
   Out Parameters : *DataSet*[datasets 4]

5. *Add an email address to the unsubscribed list*

   Operation Name: Recipients_Unsubscribe
   In Parameters : *string* email – the email address to add to unsubscribe list
   Out Parameters : *None*

6. *Get a recipient id by his email*

   Operation Name: Recipients_GetRecipientId
   In Parameters : *string* email
   Out Parameters : *int* - if the recipient was found by the email, null if none was found

7. *Get all of a recipient's actions*

   Operation Name: Recipients_ GetActionsHistory
   In Parameters : *int* – the recpient id
   Out Parameters : *DataSet*[datasets 6]

Sending

1. *Send immediate email (without sending control) - Send a single email for one recipient, the email will be send immediately and tracked for the recipients actions.*

   Operation Name: Sending_SendImmediateMessage
   In Parameters : InwiseApplicationDataProvider.*InwiseMessage*[classes 2] class
   int – the recpient id you want to attach to this recipient (this is the number that you will get back in the Datasets6 or Datasets9)
   string – email
   Out Parameters : *int* – sending result (0 - message sent to SMTP. Other number means you have an error, please see the error code by using errors_TranslateErrorCode or errors_GetErrorCodes operation

   Remarks:
   - in the InwiseMessage class you must also provice the sendid that you want
   - Synchronic sending
   - Without sending control – we will not check if the recipient has received this message (send id) before

2. *Send a single email for one recipient, the email will be send immediately and tracked for the recipients actions.*

   Remark: Sending_SendImmediatly is an old operation (was replaced *with* "Sending_SendImmediateMessage"), please do not use it

3. *Send immediate email from existing message (without sending control)*

   Operation Name: Sending_SendExistingImmediatly
   In Parameters : *int* – sendid of the message to send
   int – the recpient id you want to attach to this recipient (this is the number that you will get back in the Datasets6 or Datasets9)
   string – email
   DataSet[datasets 10] - with one row

   Out Parameters : *None*

   Remarks:
   - Synchronic sending
   - Without sending control – we will not check if the recipient has received this message (send id) before

4. *Send a message to a test group*

   Operation Name: Sending_SendTest
   In Parameters : *int* – sendid of the message to send
   int – the id of the group for which the message should be sent. This group must be a test group
   Out Parameters : *boolean* – indicating a successful send

5. *Send a message to a real group (for a message previously sent to a test group)*

   Operation Name: Sending_SendReal
   In Parameters : *int* – sendid of the message to send
   int – the id of the group for which the message should be sent
   DateTime sendTime – the time at which to send the message
   Out Parameters : *boolean* – indicating a successful send

Statistics

1. *Get all recent feedback actions* - returns all the actions added since the last call was made to this operation

   Operation Name: Statistics_GetLastFeedBack
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 6]

2. *Get action codes translation*

   Operation Name: Statistics_ TranslateActionCodes
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 11]

3. *Get bounces codes translation*

   Operation Name: Statistics_TranslateBouncesCodes
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 11]

4. *Get statistics results for a message. You will get statistic for a specific send id (how many opens, unique opens, clicks, unick clicks, sent and bounces)*

   Operation Name: Statistics_GetTotalResults
   In Parameters : *int* - sendid of the message to get statistics for
   Out Parameters : *DataSet*[datasets 13]

5. *Get feedback results for a message*

   Operation Name: Statistics_GetFeedBackForSendid
   In Parameters : *int* - sendid of the message to get statistics for
   Out Parameters : *DataSet*[datasets 6]

Errors and Codes

1. *Get Error Codes Table"*

   Operation Name: Errors_GetErrorCodes
   In Parameters : *None*
   Out Parameters : *DataSet*[datasets 11]

2. *Get Error code description*

   Operation Name: Errors_TranslateErrorCode
   In Parameters : *int* errorcode
   Out Parameters : *String – description*

Remark for all operations:

- If your request could be completed a soap exception will be thrown. The soap exception “detail” parameter will contain the error code, the “message” parameter will contain a description of the error
- All Date Time fields are in the format “**2007-05-03T13:12:41.1202621+02:00**”

InwiseApplicationDataProvider Classes

ExternalConnection. The class contains the following public fields, not all are implemented always.

Field Name	Field Type	Field Meaning
tableConId	string	
DbId	string	The database id
connectionString	string	The connection string to the database
connectionName	string	The name of the connection
groupName	string	The name of the external group
groupId	string	The id of the external group
Fields	arrayList	
tableName	string	The remote table name
dbType	int	The type of the database. 1 for MS-Sql, 0 for MS-Access
DBName	string	The name of the database

InwiseMessage. The class contains the following main public fields, not all are implemented always.

Field Name	Field Type	Field Meaning
Name	string	The name of the message
Subject	string	The subject of the message
ReplyTo	string	The reply to address
From	string	The address from which the message was sent
AttachmentFileName	string	The name of the attachment file if any
AttachmentFileData	byte array	The binary content of the attachment file if any
Body	string	The body of the message
UpdateDate	datetime	
CreationDate	datetime	The date in which the message was created
SendId	int	The sendid of the message
ContentType	string	The message content type html/text
SenderName	string	The name of the sender
MsgToName	string	The field from the custom fields settings to display in the 'to' line of the message
CharSet	string	The character set of the message

Datasets Structures

Datasets 1

The Dataset contains one table, a single row for each group. Each row contains the following columns:

Column Name	Data Type	Value
GroupName	string	The name of the group
RemoteTableName	string	The name of the table in the remote database from which the data will be retrieved
DBId	int	The id of the external database

Datasets 2

The Dataset contains one table, single row for each database. Each row contains the following columns:

Column Name	Data Type	Value
DBID	int	The id of the database
DBName	string	The name of the database
DBType	int	The type of the database – 0 for MS-Access,1 for MS-Sql
ConnectionString	string	The connection string for the database

Datasets 3

The Dataset contains one table, single row for each mapped field. Each row contains the following columns:

Column Name	Data Type	Value
localFieldId	int	The id local mapped field (can be located in the field settings menu)
remoteFieldName	string	The name of the column in the remote table for which the mapping will be created
isSlicing	int	Indicates whether the remote field has a range of closed values or is it a regular open field. Mark “1” to indicate that the field is slicing
remoteSlicingTable	string	If slicing, the name of the remote table that contains the values for that field
remoteSlicingColumn	string	If slicing, the name of the column in the remote table that contains the values for that field
remoteSlicingDescription	string	If slicing , the description of the values from the remoteSlicingColumn

Datasets 4

The Dataset contains one table, single row for each message. Each row contains the following columns:

Column Name	Data Type	Value
SendId	int	The unique sendid of the message
SendDate	Datetime	The date that the message was sent at
MessageName	*string*	The name of the message
Subject	*string*	The subject of the message

Datasets 5

The Dataset contains one table, 0 to 30 rows, with the following columns:

Column Name	Data Type	Value
name	string	The name of the fields (field1, field2, etc...)
value	object	The value of the fields. For dates it could be universal time format string or .NET DateTime object

Datasets 6

The Dataset contains one table, single row for each action:

Column Name	Data Type	Value
aid	int	the action id – this is an internal number that does not mean something
custid	int	Doesn't relevant
ActionDate	datetime	the timestamp of the action
RecipientID	int	the id of the recipient who made the action. If you use regular/local groups then it is inner id number. If you use remote groups then it is the RemoteUserId number from your own database
SendID	int	the id of the message
Action	int	The type id of the action. You have other function to get the action types For example 3 means click on link, 5 means open the mail, 11 means unsubscribe, 14 means bounce ext.
remarks	Nvarchar(250)	Additional information about the action if relevant If the action is click then it contains the link number. If the action is bounce then it contains the bounce code.

Datasets 7

The Dataset contains one table with a single row for each recipient you want to import.
Each column represents one field data about the recipient.
The column names are not fixed, each column that is to be imported should be mapped through *DataSet*[datasets 8] to the relevant field id in the system. Please note though, that mapping of the email column is required, and the fielded of the email column in system is always 0.
Example of a valid dataset:

Myemail	firstname	age
a@b.c	John	30
d@e.f	Sara	41

Datasets 8

The Dataset contains one table, single row for each field mapping:

Column Name	Data Type	Value
fieldId	int	The local id of the field
tableFieldName	int	The column name to map that field

Datasets 10

The Dataset contains one table, one row with one column for each one of the fields present in the message body and are to be replaced.
If your message contains the variables #field1# and #field3# , the data row should contain 2 columns with the names field1 and field3 and the values that are to be replaced.
For example:

field1 John
field2 Raymond

Datasets11

Column Name	Data Type	Value
Code	int	The code number
Description	string	the description of the code number

Datasets13

The Dataset returns a table with one row containing all statistics for a sendid

Column Name	Data Type	Value
opens	int	
opensUnique	int	
clicks	int	
clicksUnique	int	
sent	int	
bounces	int	

Datasets14

The Dataset contains one table, each row contains one recipient data

Column Name	Data Type	Value
Recipientid	int	the id of the recipient
Email	int	the email of the recipient
Registrationdate	datetime	the registration date of the recipient
Field1…field20	string	personal information about the recipient
Field20…field25	datetime	personal information about the recipient
Field26…field30	int	personal information about the recipient
Ce	int	a number indicating whether the recipient's email is syntactically correct
Confirm_regist	int	a number indicating whether the recipient has confirmed his registration if a one was required

Datasets15

The Dataset contains one table, each row contains one group information

Column Name	Data Type	Value
Name	string	the name of the group
Id	int	the id of the group

Datasets 16

The Dataset contains one table, each row contains one message distribution in the sending queue

Column Name	Data Type	Value
Id	string	id of the message distribution
ai	int	the priority of the message distribution
sendid	int	the id of the message
msgsubject	string	the message subject
confirmed	int	a number indicating whether the distribution has been confirmed and waiting to be sent
name	string	name of the group the message is being sent to
msgAttachFileName	string	the name of an attached file in the message
TimeToStartDistribution	datetime	the time at which the distribution process should start
Queryid	int	the id of the query the message is being sent to
Groupid	int	the id of the group the message is being sent to

Datasets17

The Dataset contains one table, each row contains one import information

Column Name	Data Type	Value
Id	int	the id of the import
Importstarted	int	a number indication whether the import process has started already
fileNameWithoutPath	string	the name of file to import
fieldsmeaning	string	the mapping of the file column names to the custom fields defined in the system
bmark	int	the current row in the file the imported process has reached to
error_import	string	if an error occurred in the import process, this column should be an indication for the reason
confirmed	int	a number indicating whethe the import has been confirmed and waiting for start
isBuld	int	a number indicating whethe the import is will made using a bulk insert
totalrecords	int	the initial number of rows encountered in the file for importing
end_error	int	a number indicating whethe an error occurred in the proccess
ai	int	the priority of the import
name	string	the name of the group the recipients will be imported to
MailingListToImport	int	the id of the group the recipinets will be imported to