

Synthesys Dialler Web Service API

Version 2.4

VERSION CONTROL

Version	Author	Comments	Revision Date
---------	--------	----------	---------------

2.0	Luke Harris	New version	3 rd March 2014
2.1	Vladimir Wegner	Adding new methods	29 th March 2016
2.2	Craig Butters	Reformat and review	20 th February 2018
2.3	Paul Keyes	Added Extra CreateUpdate clarifications and new TelephoneName parameter	9 th October 2020
2.4	Max	Added CLIToPresent & DialPrefix	15 th September 2022

TABLE OF CONTENTS

Version Control	1
Overview	4
Calling the API	4
Security	4
Queuing Manipulation APIs	4
CreateQueuedRecord(s)	4
DeleteQueuedRecord	7
DeleteQueuedRecords	8
UpdateQueuedRecord	8
UpdateQueuedRecordWithPriority	8
UpdateQueuedRecords	9
CreateUpdateQueuedRecord	9
CreateUpdateQueuedRecordWithPriority	10
Describe and Search functions	11
Queue States and Meanings	14
Troubleshooting	14

OVERVIEW

The Synthesys Dialler Web Service provides an API into the Synthesys Dialler subsystem. The current version allows manipulation of the Queue table; future versions will add fuller integration with the Predictive Dialler and also access to the CTI facilities of Synthesys.

This service can be used in conjunction with the CRM Web Service to remove the need for the CRMWebTransfer service.

Note: Any Noetica API can only be used when authorized by Noetica. Additional training may need to be provided to 3rd party developers who wish to access the Noetica APIs. Contact your account manager or Noetica contact for full details.

CALLING THE API

The API is presented on the URL <http://<synthesysserver>DiallerWebService>, with a WSDL available for binding at URL <http://<synthesysserver>DiallerWebService?wsdl> (this can be turned off if required for security reasons). We can also supply the WSDL as a file if required.

SECURITY

All Web Services take a username/password parameter, which is that of an authenticated Synthesys user which can be set up using the Synthesys tools.

QUEUING MANIPULATION APIS

CREATEQUEUEDRECORD

Use this method to create a single record. To create multiple records, use **CreateQueuedRecords** instead. For more flexibility, consider using **CreateUpdateQueuedRecord**

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
OBCampaignId	Int	Phoenix_obcampaigns
CustomerPrefix	String	
CustomerId	String	
State	Int	See below for valid states
TimesCalled	Int	Can be more than zero if used together with delete for a move operation.
AgentId	Int	
Retry_Time	String	Can be null if not required.
Latest_Time	String	Can be null if not required.
Telephone	String	
SortOrder	String	See below.
TelephoneName	String	See below
CLIToPresent	String	See below
DialPrefix	String	See below

Creates a new record in the queue table, writes a record to the history table using information from the username.

SortOrder is to support the new optional SortOrder column in the Phoenix_Queue table, which is new for V4.1 of Synthesys. For this release of the API, SortOrder must be blank.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

If the state selected is Instant Call Back then the PD will also be notified and the call given to the next available agent without the need for the PD to fetch the record from the database.

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

CLIToPresent and DialPrefix will go into their new columns in Phoenix_Queue table introduced with 22.03

CreateQueuedRecords

Parameter	Data Type	Notes
Records	List of QueuedRecord	See the parameter data type description below.

Field Name	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
OBCampaignId	Int	Phoenix_obcampaigns
CustomerPrefix	String	
CustomerId	String	
State	Int	See below for valid states
TimesCalled	Int	Can be more than zero if used together with delete for a move operation.
AgentId	Int	
Retry_Time	String	Can be null if not required.
Latest_Time	String	Can be null if not required.
Telephone	String	
SortOrder	String	
TelephoneName	String	

Creates new records in the queue table, writes records to the history table using information from the parameter for each record.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

The method returns a list of record QIDs

CreateQueuedRecordWithLeadID

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
OBCampaignId	Int	Phoenix_obcampaigns
CustomerPrefix	String	
CustomerId	String	
State	Int	See below for valid states
TimesCalled	Int	Can be more than zero if used together with delete for a move operation.
AgentId	Int	
Retry_Time	String	Can be null if not required.
Latest_Time	String	Can be null if not required.
Telephone	String	
SortOrder	String	
LeadID	Int	
TelephoneName	String	
CLIToPresent	String	See below
DialPrefix	String	See below

Creates a new records in the queue table and lead tracking table, writes arecord to the history table using information from the Username.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

CLIToPresent and DialPrefix will go into their new columns in Phoenix_Queue table introduced with 22.03

The method returns QID of the newly added record

CreateQueuedRecordWithLeadIDAndPriority

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
OBCampaignId	Int	Phoenix_obcampaigns
CustomerPrefix	String	
CustomerId	String	
State	Int	See below for valid states

TimesCalled	Int	Can be more than zero if used together with delete for a move operation.
AgentId	Int	
Retry_Time	String	Can be null if not required.
Latest_Time	String	Can be null if not required.
Telephone	String	
SortOrder	String	
LeadID	Int	
Priority	Int	See below
TelephoneName	String	
CLIToPresent	String	See below
DialPrefix	String	See below

Creates a new records in the queue table and lead tracking table, writes arecord to the history table using information from the Username.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

Actual order of the records is defined by OB Campaign's property PriorityOrder

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

CLIToPresent and DialPrefix will go into their new columns in Phoenix_Queue table introduced with 22.03

The method returns QID of the newly added record

CreateQueuedRecordsCLI

Parameter	Data Type	Notes
Records	List of QueuedRecordCLI inherited from QueuedRecord class	See the parameter data type description below.

Field Name	Data Type	Notes
QueuedRecord properties inherited from QueuedRecord class		
CLIToPresent	String	
DialPrefix	String	

Creates new records in the queue table, writes records to the history table using information from the parameter for each record.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

CLIToPresent and DialPrefix will go into their new columns in Phoenix_Queue table introduced with 22.03

The method returns a list of record QIDs

DELETEQUEUEDRECORD

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
QueueId	Int	Item to delete.

Deletes a record from the queue table, writes a record to the history table using information from the LogonToken.

DELETEQUEUEDRECORDS

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
QueueId	String	Comma separated list of Ids, limited as for SearchQueuedRecords. An exception is returned if too many are specified.

This is identical to the existing DeleteQueueRecord function, except with a string of Queue Ids instead of a single Id.

UPDATEQUEUEDRECORD

Use this method to update a single record. To update multiple records, use **UpdateQueuedRecords** instead. For more flexibility, consider using method **CreateUpdateQueuedRecord**

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
QueueId	Int	
NewState	Int	
RetryTime	Null or datetime	
LatestTime	Null or datetime	
AgentId	Int	

UPDATEQUEUEDRECORDWITHPRIORITY

For more flexibility, consider using method **CreateUpdateQueuedRecordWithPriority**

Parameter	Data Type	Notes
-----------	-----------	-------

Username	String	See security above.
Password	String	See security above.
QueueID	Int	
NewState	Int	
RetryTime	Null or datetime	
LatestTime	Null or datetime	
AgentId	Int	
Priority	Int	

The method is identical to UpdateQueueRecord function, except it allows to update the record's priority

UPDATEQUEUEDRECORDS

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
QueueIDs	String	Comma separated list of Ids, limited as for SearchQueuedRecords. An exception is returned if too many are specified.
NewState	Int	
RetryTime	Null or datetime	
LatestTime	Null or datetime	
AgentId	Int	

This is identical to the existing UpdateQueueRecord function, except with a string of Queue Ids instead of a single Id.

CREATEUPDATEQUEUEDRECORD

Parameter	Data Type	Notes	Used In
Username	String	See security above.	Create & Update
Password	String	See security above.	Create & Update
OBCampaignId	Int	Phoenix_obcampaigns	Create
CustomerPrefix	String		Create
CustomerId	String		Create
QueueID	Int	See below	See below
State	Int	See below for valid states	Create & Update
TimesCalled	Int	Can be more than zero if used together with delete for a move operation.	Create
AgentId	Int		Create & Update
Retry_Time	String	Can be null if not required.	Create & Update
Latest_Time	String	Can be null if not required.	Create & Update
Telephone	String		Create

SortOrder	String		Create
TelephoneName	String	See below	Create
CLIToPresent	String	See below	Create
DialPrefix	String	See below	Create

This method creates a new record in the queue table or updates an existing record. If the parameter QueueID is 0, a new record will be created. If the QueueID is not 0, the record with the QueueID will be updated.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

CLIToPresent is new CLIToPresent column in Phoenix_Queue table appeared in 22.03

DialPrefix is new DialPrefix column in Phoenix_Queue table appeared in 22.03

CREATEUPDATEQUEUEDRECORDWITHPRIORITY

Parameter	Data Type	Notes	Used In
Username	String	See security above.	Create & Update
Password	String	See security above.	Create & Update
OBCampaignId	Int	Phoenix_obcampaigns	Create
CustomerPrefix	String		Create
CustomerId	String		Create
QueueID	Int	See below	See below
State	Int	See below for valid states	Create & Update
TimesCalled	Int	Can be more than zero if used together with delete for a move operation.	Create
AgentId	Int		Create & Update
Retry_Time	String	Can be null if not required.	Create & Update
Latest_Time	String	Can be null if not required.	Create & Update
Telephone	String		Create
SortOrder	String		Create
Priority	Int		Create & Update
TelephoneName	String	See below	Create
CLIToPresent	String	See below	Create
DialPrefix	String	See below	Create

This method creates a new record in the queue table or updates an existing record. If the parameter QueueID is 0, a new record will be created. If the QueueID is not 0, the record with the QueueID will be updated.

Valid states for this operation are Queued (1), Sleeping (6), Rescheduled (2), or Instant Call Back (14)

TelephoneName can be omitted, but if it is sent it must be a valid Telephone Field Name in the CustomerPrefix entity

CLIToPresent is new CLIToPresent column in Phoenix_Queue table appeared in 22.03

DialPrefix is new DialPrefix column in Phoenix_Queue table appeared in 22.03

DESCRIBE AND SEARCH FUNCTIONS

DescribeQueuedRecord

This takes three parameters, username, password and an integer called 'Id', which is the queue id.

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
Id	Int	

It returns an XML string, describing the contents of the queued record, or it will return a blank string if no record matches that queue id.

Returned XML is this. All field names and values are returned as properties. Each property has a name corresponding to a column name, and a value of the field:

```
"<Properties><Property Name="ID" Value="1" /><Property Name="OBCampaign_ID" Value="1"/><Property Name="Customer_ID" Value="Elect_1" />...etc...</Properties>"
```

Note. Retry Time and Latest Time are returned in App Server Time Zone, that is, exactly the values

SearchQueuedRecords

This takes three parameters, username, password and a string called 'SearchString'.

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
SearchString	String	See section Search String

Search String

The search string is formatted like a URL encoded search string, with fields matching columns in Phoenix_Queue, so for example

"Customer_ID=Elect_1&OBCampaign_ID=1" (All queued records for Elect_1 in list id 1)
or "Telephone=012345678&State=1" (All queued records for Telephone 012345678)
or "OBCampaign_ID=1&State=3" (all attention calls for List with Id 1)

The string is URL encoded, so for examples spaces will be encoded as %20%.

The return is string containing a comma separated list of matching IDs from Phoenix_Queue. If no matching records are found, the string will be empty. The number of matches will be limited by the setting in Synthesys.INF, which defaults to 100 if not found:

[DiallerWebService]

MaximumRecords=100

Warning: Setting this value too high can produce performance problems for the predictive dialler if use of the dialler web service puts too much load on the queue table.

Complex Queries

The syntax above give 'AND' queries. We also allow the equivalent of IN syntax:

Parameter	Data Type	Notes
Username	String	

CallRecording	Bool	
---------------	------	--

Customer_ID=ELECT_1&State=(1,2) is equivalent to the SQL statement
Customer_Id='ELECT_1' AND State IN (1,2)

We also allow NOT by using the ! operator Customer_ID=ELECT_1&State=!3 is equivalent to the SQL statement Customer_Id='ELECT_1' AND State <> 3

And finally we allow LIKE by automatically spotting queries with %, so for example Telephone=0783% is equivalent to Telephone LIKE '0783%'

AddToDoNotCallTable

Parameter	Data Type	Notes
Username	String	See security above.
Password	String	See security above.
Telephone	String	
Expires	DateTime	
Flags	String	
Scope	String	
ScopeID	String	
Source	String	
Info	String	

This method add a new record to the Do Not Call table. All parameters are described in a separate document for Advanced Do Not Call Functionality

StopCallRecording

Agent	Bool	
Customer	Bool	

ResumeCallRecording

Parameter	Data Type	Notes
Username	String	
CallRecording	Bool	
Agent	Bool	
Customer	Bool	

QUEUE STATES AND MEANINGS

0	Done
1	Queued
2	Rescheduled
3	Attention
4	Never call
5	Running
6	Sleeping (from Queued)
7	Sleeping (from Scheduled)
8	Outbound Delete
9	Wrong Number
10	Unobtainable
11	Predictive Call Running
12	Recycling Done
13	Predictive Running
14	Instant Call Back
15	Moved

TROUBLESHOOTING

Some Dialler Web Service actions for example Update and Delete and handled asynchronously by the Predictive Dialler. This means that faults can't be returned by the web service, instead the Predictive Dialler log should be consulted. The following messages might be logged for example:

*Ignoring OBManager Update Request for Queue ID **ID** as it has Pending Updates or
Ignoring OBManager Update Request for Queue ID **ID** as it has Pending Delete Reque*