DemandCenter Integration API

Version 1.0.6
August 15, 2016

eTrigue Corporation
6399 San Ignacio Avenue
Second Floor
San Jose, CA 95119
www.etrigue.com

800.858.8500 toll-free
408.490.2901 fax

Contents

Introduction

The eTrigue integration API allows customers to access prospect information stored in eTrigue, facilitating custom integration to CRM and other systems. The API uses the well-known SOAP protocol and is easily interfaced from most systems and technology deployed in current distributed environments. The API supports retrieving, creating, and updating prospects from the eTrigue system as well as access to prospect activity data via the eTrigue activity links.

This interface is provided only to registered eTrigue customers, and is considered eTrigue proprietary. Only customer employees may use or have access to this document as it is protected under copyright law and non-disclosure agreements with eTrigue Corp. In order to use this API customer must have an active eTrigue account.

Getting Access

Customers that desire access to the API must request activation of the interface for a specific user account. We recommend that a set of credentials be created specifically for accessing the API. Once the account is activated and a specific user enabled we will provide test and production URL's to use for accessing the corresponding environment. A test database will be populated with some sample data to get development started. Each test environment is restricted to a single customer access so no one else will be able to access any data that is loaded. Approximately 1 month after you have gone live with the production database integration the test database will be removed.

For security purposes all calls to the API must use https, no http access is allowed.

API Calls

The following sections describe the structure required for each API call and the data structure returned by the call.

All references to timestamp parameters require the format YYYY-MM-DD HH:MM:SS

Login - Login()

Before any calls can be made, a login call must be made to receive a secure token to use in all other calls. The token received will remain valid until there is a five minute period of inactivity. After that, any attempts to use the token will result in an error. To resume making calls to the API, a new token must be obtained (i.e., login again).

Parameter	Description
Login	The login name for the account authorized by eTrigue support to use the API
Password	The account password

Request Login

ESPAccount = Login(string username, string password);

Response Login

On success, the call returns the LoginResult object which contains

```
public class ESPAccount
{
  string UserName
  string Password
  string SessID
  int UserID
}
```

For example;

```
stdClass Object {
  [Password] => xxxxxxxxxx
  [SessID] => 8d05211ddd44557415de57f2f946ef53
  [UserName] => usernamehere
  [UserID] => 12
 }
```

Request Prospect Field List – reqProspectFieldList()

To provide the correct mapping to the eTrigue system, customers must map the CRM system field names to the field names used within the eTrigue system. This call returns the list of field names to use when providing data to the API; these are the names the API is expecting when providing data. The list of fields includes any User Defined Fields (UDFs) that were added to the customer's eTrigue account.

Parameter	Description
Session Object	The session object returned from the login call

Request Prospect Field List

```
resProspectFieldList = reqProspectFieldList(ESPAccount esp);
```

Response Prospect Field List

On success, the call returns the prospectFieldList object which contains an array of:

```
public class resProspectFieldList
{
  string FieldName
  string FieldType
  string FieldSize
  UDFvalues MultiSelectValues
}

public class UDFvalues
{
  public string UDFFieldId
  public string UDFFieldValue
}
```

For example;

```
        [resEtrigueFieldList] => Array (
                [0] => stdClass Object {
                        [FieldName] => eTrigueID
                        [FieldType] => int
                        [MultiSelectValues] =>
                }
                [1] => stdClass Object {
                        [FieldName] => emailAddress
                        [FieldType] => varchar
                        [MultiSelectValues] =>
                }
        …
        [20] => stdClass Object {
            [FieldName] => UDF_Functional Level
            [FieldType] => varchar
            [MultiSelectValues] => stdClass Object {
                [UDFvalues] => Array (
                    [0] => stdClass Object {
                        [UDFFieldValue] => C-Level
                    }
                    [1] => stdClass Object
                        [UDFFieldValue] => Sales
                    }
                    [2] => stdClass Object {
                        [UDFFieldValue] => Marketing
                    }
        …
    )
```

The MultiSelectValues array will be included for fields that provide multi-select values. The array lists the values that are allowed for the field.

Note: Any changes to the eTrigue UDF fields will need to be accounted for in the integration that uses the API.

Refer to Appendix B for standard field schema definitions.

Request Prospect Ids – reqProspectIds()

The call returns a list of all eTrigue prospect Id's along with their matching external Id, if present.

Parameter	Description
Session Object	The session object returned from the login call
Last Modified Time-stamp	Return all prospect IDs that have been modified since this time-stamp
List Index	This optional parameter is used to obtain the next block of IDs. If the parameter is omitted, the first 500 IDs are returned; if the value is the last index (i.e., eTrigue Id) from the previous response, the next 500 entries are returned. The index is used to advance through the list until all IDs are returned.

Request Prospect Id's

```
resEtrigueIds = reqProspectIds(ESPAccount esp, string lastmodifieddate, int listindex);
```

Response Prospect Id's

On success, the call returns an array of eTrigueID objects:

```
public class resEtrigueIds
{
  string eTrigueID
  string ExternalID
}
```

For example;

```
        Array (
          [0] => stdClass Object {
               [ExternalID] =>
               [eTrigueID] => 110321
                 }

          [1] => stdClass Object {
               [ExternalID] => 0234567
               [eTrigueID] => 110322
                }
                ...
        )
```

The object will always return the eTrigue ID. It will also provide the external ID (ID for the matching record in the customer's CRM system) if the records have been related. Depending on how prospects are managed and integrated with the customer's CRM system, this may or may not be populated.

Note: Please review the de-duplication processing rules used by eTrigue in Appendix A. This will affect when and how records are mapped from the customer's CRM system to eTrigue.

Request Prospect Updates – reqPropectUpdates()

The call returns a list of all prospect records updated since the timestamp provided, if more than 500 records are supplied additional calls will return more records based on the list index parameter.

Parameter	Description
Session Object	The session object returned from the login call
Last Modified Time-stamp	Return all prospects that have been modified since this timestamp.
List Index	This optional parameter is used to obtain the next block of update. If the parameter is omitted, the first 500 updates are returned; if the value is the last index (i.e., eTrigue Id) from the previous response, the next 500 entries are returned. The index is used to advance through the list until all updates/prospects are returned.

Request Prospect Updates

```
resProspectData = reqProspectUpdates(ESPAccount esp, string lastmodifieddate, int listindex);
```

Response Prospect Updates

On success, the response returns an array of the resProspectData object.

```
public class resProspectData
{
   string eTrigueID
   string FirstName
   string LastName
   string Companyname
   string Profile
   string EmailAddress
   string Title
   string Salutation
   string Address
   string Address2
   string Address
   string City
   string State
   string Zipcode
   string Country
   string Phone
```

```
  string Phone2
  string Fax
  string URL
  string Memo
  string eTriguelink
  string Source
  string Externalid
  string Industry
  string Segment
  string Territory
  string AccountType
  string CurrencyType
  string CurrencyValue
  string CrmOwner
  string ProspectOwner
  string Stage
  string Status
  string Gender
  UDFdata[] UDFData
}

public class UDFdata
{
  string udfname
  string udfvalue
  string UDFName
  string UDFValue
}
```

For example;

```
        Array (
                [0] => stdClass Object {
                        [Address] => PO BOX 16560
                        [Address2] =>
                        [Address3] =>
                        [City] => ALBUQUERQUE
                        [Companyname] => ACME
                        [Country] => US
                        [EmailAddress] => js@gmail.com
                        [Externalid] =>
                        [Fax] =>
                        [FirstName] => John
                        [LastName] => Smith
                        [Memo] =>
                        [Phone] =>
                        [Phone2] =>
                        [Profile] => List_UnknownRev_Manager_8-07
                        [Salutation] =>
                        [State] => NM
                        [Title] => CEO
                        [UDFData] => stdClass Object {
                            [UDFdata] => Array (
                                [0] => stdClass Object {
                                    [UDFName] => Functional Level
                                    [UDFValue] => "Top,"
                                }
                                [1] => stdClass Object {
                                    [UDFName] => SIC 1
                                    [UDFValue] => 0
                                }
                                …
                            )
                        }
                        [URL] => www.acme.com
                        [Zipcode] => 87191
                        [eTrigueID] => 110321
                        [eTriguelink] =>
                }
                …
        )
```

Request Prospect Updates By Id – reqPropectUpdatesById()

The call returns a list of all prospect records based on the list of eTrigue Ids provided as input. If more than 500 records will be returned, additional calls will return more records based on the list index parameter.

Parameter	Description
Session Object	The session object returned from the login call
Array of eTrigueIDs	An array of the eTrigueIds objects. Refer to reqProspectId's call.
List Index	This optional parameter is used to obtain the next block of updates if the parameter is omitted, the first 500 updates are returned; if the value is the last index (i.e., eTrigue Id) from the previous response, the next 500 entries are returned. The index is used to advance through the list until all updates/prospects are returned.

Request Prospect Updates By Id

```
resProspectData = reqProspectUpdatesById(ESPAccount esp, array resEtrigueIds, int listindex);
public class resEtrigueIds
{
  string eTrigueID
  string ExternalID
}
```

Response Prospect Updates By Id

On success, the response returns an array of the resProspectData object:

```
        Array (
                [0] => stdClass Object {
                        [Address] => PO BOX 16560
                        [Address2] =>
                        [Address3] =>
                        [City] => ALBUQUERQUE
                        [Companyname] => ACME
                        [Country] => US
                        [EmailAddress] => js@gmail.com
                        [Externalid] =>
                        [Fax] =>
                        [FirstName] => John
                        [LastName] => Smith
                        [Memo] =>
```

```
            [Phone] =>
            [Phone2] =>
            [Profile] => List_UnknownRev_Manager_8-07
            [Salutation] =>
            [State] => NM
            [Title] => CEO
            [UDFData] => stdClass Object {
                [UDFdata] => Array (
                    [0] => stdClass Object {
                        [UDFName] => Functional Level
                        [UDFValue] => "Top,"
                    }
                    [1] => stdClass Object {
                        [UDFName] => SIC 1
                        [UDFValue] => 0
                    }
                    …
                )
            }
            [URL] => www.acme.com
            [Zipcode] => 87191
            [eTrigueID] => 110321
            [eTriguelink] =>
        }
        …
    )
```

Request Prospect Upsert – reqProspectUpsert()

This call updates or inserts (i.e., upsert) records into the eTrigue system. The decision to insert or update is based on the de-duplication processing logic described in Appendix A.

Parameter	Description
Session Object	The session object returned from the login call
Array of object resProspectData	Array of resProspectData objects; the objects are updated or inserted depending on whether matching records are already in the system. There is a limit of 500 prospects records per call.

Request Prospect Upsert

resResult = reqProspectUpsert(ESPAccount esp, resProspectData[] custrecs);

Response Prospect Upsert

On success, the response returns the resUpsertResult object. The array has a one-to-one entry for each record that was received in the Upsert request indicating whether it was successfully inserted, updated, or encountered an error. If the record matches an existing record based on the de-duplication processing, the eTrigue ID for that record will be returned.

```
public class resResult
{
  public string etrigueID
  public bool Success
  public string ErrorMessage
}
```

For example;

```
      Array (
            [0] => stdClass Object {
                  [ErrorMessage] =>
                  [Operation] => UPDATE
                  [Success] => True
                  [etrigueID] => 12776
            }
            [1] => stdClass Object {
                  [ErrorMessage] =>
                  [Operation] => INSERT
                  [Success] => True
                  [etrigueID] => 13777
            }
      )
```

Request Prospect Statistics – reqProspectStatistics()

This call returns the statistic data for all prospects with statistics updated since the time stamp provided. This data is provided separately as it typically changes more frequently than other prospect data.

Parameter	Description
Session Object	The session object returned from the login call
Last Modified Time-stamp	Return the statistics for the prospects modified since the last timestamp.
List Index	This optional parameter is used to obtain the next block of updates. If the parameter is omitted, the first 500 updates are returned; if the value is the last index from the previous response, the next 500 entries are returned. The index is used to advance through the list until all prospect statistics are returned.

Request Prospect Statistics –by last modified date

```
resProspectStats = reqProspectStatistics(ESPAccount esp, string lastmodifieddate, int listindex);
```

Response Prospect Statistics – by last modified date

The response returns the object resProspectStatistics with an array of the statistics object.

```
public class resProspectStats
{
  public string eTrigueID
  public int VisitCount
  public int PageCount
  public int LengthOfStay
  public string DateAdded
  public string LastVisitDate
  public string ExternalID
}
```

For example;

```
        Array (
                [0] => stdClass Object (
                        [DateAdded] => 2010-06-26 22:41:43
                        [ExternalID] => abc123
                        [LastVisitDate] => 2008-08-15 17:28:30
                        [LengthOfStay] => 12
                        [PageCount] => 5
                        [VisitCount] => 2
                        [eTrigueID] => 110321
                }
                [1] => stdClass Object {
                        [DateAdded] => 2010-07-26 22:41:43
                        [ExternalID] => abc124
                        [LastVisitDate] => 2008-08-17 17:28:44
                        [LengthOfStay] => 102
                        [PageCount] => 8
                        [VisitCount] => 4
                        [eTrigueID] => 110322
                }
        …
        )
```

Request Prospect Statistics By Id – reqProspectStatisticsById()

This call returns the statistic data for the prospects, based on the prospect Id's provided as input to the call. This data is provided separately as it typically changes more frequently than other prospect data. If more than 500 records will be returned, additional calls will return more records based on the list index parameter.

Parameter	Description
Session Object	The session object returned from the login call
Array of eTrigueIDs	An array of the eTrigueIds objects. Refer to reqProspectId's call.
List Index	This optional parameter is used to obtain the next block of updates. If the parameter is omitted, the first 500 updates are returned; if the value is the last index from the previous response, the next 500 entries are returned. The index is used to advance through the list until all prospect statistics are returned.

Request Prospect Statistics By Id

```
resProspectStatDetails = reqProspectStatDetails(ESPAccount esp, array resEtrigueIds, int listindex);
public class resEtrigueIds
{
  string eTrigueID
  string ExternalID
}
```

Response Prospect Statistics By Id

The response returns the object resProspectStatisticsById with an array of the statistics object.

```
public class resProspectStatDetails
{
  public string eTrigueID
  public resVisits[] Visits
  public resEmails[] Emails
  public resFormFills[] FormFills
  public resDownload[] Download
}

public class resVisits
{
  public string TimeStamp
  public string URL
}

public class resEmails
{
  public string TimeStamp
  public string Name
  public string Status
}

public class resFormFills
{
  public string TimeStamp
  public string NameOfForm
}

public class resDownload
{
  public string TimeStamp
  public string DownloadFile
}
```

For example;

```
        Array (
                [0] => stdClass Object (
                        [DateAdded] => 2010-06-26 22:41:43
                        [ExternalID] => abc123
                        [LastVisitDate] => 2008-08-15 17:28:30
                        [LengthOfStay] => 12
                        [PageCount] => 5
                        [VisitCount] => 2
                        [eTrigueID] => 110321
                }
                [1] => stdClass Object {
                        [DateAdded] => 2010-07-26 22:41:43
                        [ExternalID] => abc124
                        [LastVisitDate] => 2008-08-17 17:28:44
                        [LengthOfStay] => 102
                        [PageCount] => 8
                        [VisitCount] => 4
                        [eTrigueID] => 110322
                }
        )
```

Request Prospect Statistic Details – reqProspectStatDetails()

This call returns the statistic details for all prospects with statistics updated since the time stamp provided. This data is provided separately as it typically changes more frequently than other prospect data. The information returned by this call:

- Email: timestamp it was sent, status (sent, opened, bounced)
- Form Fill: timestamp the form was submitted, form name used for the submission
- Visit: timestamp, the URL of the page visited.
- Download: timestamp, URL of the page downloaded

Parameter	Description
Session Object	The session object returned from the login call
Last Modified Time-stamp	Return the statistics for the prospects modified since the last timestamp.
List Index	This optional parameter is used to obtain the next block of updates. If the parameter is omitted, the first 500 updates are returned; if the value is the last index from the previous response, the next 500 entries are returned. The index is used to advance through the list until all prospect statistics are returned.

Request Prospect Statistics Details – by last modified date

resProspectStatDetails = reqProspectStatDetails(ESPAccount esp, string lastmodifieddate, int listindex);

Response Prospect Statistics Details – By last modified date

The response returns the object with an array resProspectStatDetails .

```
public class resProspectStatDetails
{
  public string eTrigueID
  public resVisits[] Visits
  public resEmails[] Emails
  public resFormFills[] FormFills
  public resDownload[] Download
}

public class resVisits
{
  public string TimeStamp
  public string URL
}

public class resEmails
{
  public string TimeStamp
  public string Name
  public string Status
}

public class resFormFills
{
  public string TimeStamp
  public string NameOfForm
}

public class resDownload
{
  public string TimeStamp
  public string DownloadFile
}
```

For example;

```
        resProspectStatDetails => Array (
                [0] => stdClass Object (
                        [Download] =>
                                [resDownload] => Array
                                        [0] => stdClass Object
                                                [DownloadFile] => aaaaaaTest
                                                [Timestamp] => 2011-06-07 15:54:27
                                        [1] => stcdClass Object
                                                [DownloadFile] => bbbbbTest
                                                [Timestamp] => 2011-06-09 10:24:17

                        [Emails] => stdClass Object
                                [resEmails] => Array
                                        [0] => stdClass Object
                                                [Name] => aaaaaaTest
                                                [Status] => Delivered
                                                [Timestamp] => 2011-06-07 15:54:27
                                        [1] => stcdClass Object
                                                [Name] => bbbbbTest
                                                [Status] => Bounced
                        [Timestamp] => 2011-06-09 10:24:17
            [FormFills] =>
                [resFormFills] => Array
                    [0] => stdClass Object
                        [NameOfForm] => Reply on visit
                        [Timestamp] => 2011-06-07 15:54:27
                    [1] => stcdClass Object
                        [NameOfForm] => DC Lunch Learn
                        [Timestamp] => 2011-06-09 10:24:17

            [Visits] => stdClass Object
                [resVisits]  => Array
                    [0] => stdClass Object
                        [Timestamp] => 2011-06-07 15:54:27
                        [URL] => http://www.etrigue.com
                    [1] => stcdClass Object
                        [Timestamp] => 2011-06-08 13:14:03
                        [URL] => http://www.etrigue.com
            [eTrigueID] => 110321
    )
    …
}
```

Request Prospect Statistic Details By Id – reqProspectStatDetailsById()

This call returns the statistic details for all prospects that are passed into the call as input. This data is provided separately as it typically changes more frequently than other prospect data. If more than 500 records will be returned, additional calls will return more records based on the list index parameter. The information returned by this call:

- Email: timestamp it was sent, status (sent, opened, bounced)
- Form Fill: timestamp the form was submitted, form name used for the submission
- Visit: timestamp, the URL of the page visited.
- Download: timestamp, URL of the page downloaded

Parameter	Description
Session Object	The session object returned from the login call
Array of eTrigueIDs	An array of the eTrigueIds objects. Refer to reqProspectIds call.
List Index	This optional parameter is used to obtain the next block of updates. If the parameter is omitted, the first 500 updates are returned; if the value is the last index from the previous response, the next 500 entries are returned. The index is used to advance through the list until all prospect statistics are returned.

Request Prospect Statistics Details By Id – Organized by list of Id's

```
resEtrigueIds = reqProspectStatDetailsById(ESPAccount esp, array resEtrigueIds, int listindex);
public class resEtrigueIds
{
  string eTrigueID
  string ExternalID
}
```

Response Prospect Statistics Details By Id – Organized by list of Id's

The response returns the object with an array resProspectStatDetailsById.

```
resProspectStatDetailsById => Array (
        [0] => stdClass Object (
                [Download] =>
                        [resDownload] => Array
                                [0] => stdClass Object
                                        [DownloadFile] => aaaaaaTest
                                        [Timestamp] => 2011-06-07 15:54:27
                                [1] => stcdClass Object
                                        [DownloadFile] => bbbbbTest
                                        [Timestamp] => 2011-06-09 10:24:17
```

```
            [Emails] => stdClass Object
                [resEmails] => Array
                    [0] => stdClass Object
                        [Name] => aaaaaaTest
                        [Status] => Delivered
                        [Timestamp] => 2011-06-07 15:54:27
                    [1] => stcdClass Object
                        [Name] => bbbbbTest
                        [Status] => Bounced
                        [Timestamp] => 2011-06-09 10:24:17
            [FormFills] =>
                [resFormFills] => Array
                    [0] => stdClass Object
                        [NameOfForm] => Reply on visit
                        [Timestamp] => 2011-06-07 15:54:27
                    [1] => stcdClass Object
                        [NameOfForm] => DC Lunch Learn
                        [Timestamp] => 2011-06-09 10:24:17

            [Visits] => stdClass Object
                [resVisits]  => Array
                    [0] => stdClass Object
                        [Timestamp] => 2011-06-07 15:54:27
                        [URL] => http://www.etrigue.com
                    [1] => stcdClass Object
                        [Timestamp] => 2011-06-08 13:14:03
                        [URL] => http://www.etrigue.com
            [eTrigueID] => 110321
    )
    …
}
```

Request Prospect Merge – reqProspectMerge()

This call merges two prospects, regardless of any data dis-similarities. This allows the calling service to merge two records combining the activity history of the duplicate prospect to the master and deleting the duplicate record. NOTE: There is no ability to reverse the merge command.

When two prospects are merged several data operations occur as a transaction:

- All activity history for the duplicate is moved to the master.
- Any categories that the duplicate is related to are added to the categories the master is related to. The result is the master is related to the superset of categories between the two prospect records.
- Any unsubscribe settings the duplicate has are applied to the master.
- If either the duplicate or the master is opted-out the master remains opted-out.
- Any fields in the master that are empty are populated by the duplicate, if the duplicate has values for those fields. No attempt is made to combine or merge any field values.
- The duplicate is deleted

Parameter	Description
Session Object	The session object returned from the login call
eTrigue ID	The original eTrigue ID, this is retained as the prospect of record.
eTrigue ID	The duplicate eTrigue ID, this will be deleted. NOTE: There is no reversing this command once executed.

Response Prospect Merge

bool reqProspectMerge(ESPAccount esp, string orgID, string dupID);

Response Prospect Merge

The response returns bool TRUE = success, FALSE = error with a message.

Appendix A - De-Duplication Processing

De-duplication processing is an integral part of eTrigue, with the goal being to eliminate redundant prospect data and provide a cleaner source of leads and marketing information. eTrigue attempts to map any new prospects using the following rules:

1. Any Upsert (insert or update) records provided to the API that have an eTrigue Id will use that Id to match an eTrigue record. If there is no match to the eTrigue ID provided, the record will not be processed and an error will be returned in the response for the record. If no eTrigue Id is provided then the next rule is checked.

2. Attempt to match the external ID that is provided.

   ```
   If the external ID matches an existing external ID
       this is an existing record perform an update
   If the external ID matches multiple records, also try to match on email address, first name, and
       last name. Use the first record that matches, if no records match use the first record.
   Else
       continue to the next rule.
   ```

3. Attempt to match the email address, first name, and last name that are provided.

   ```
   If the email address, first name, last name matches a single existing record perform an update
   Else if there are multiple matches then use the first one
   Else
       continue to the next rule.
   ```

4. Attempt to match the first name, last name, and company that are provided.

   ```
   If the first name, last name, and company match an existing prospect's values
       this is a dup and we should use the existing record in eTrigue.
   Else
       create a new prospect record.
   ```

Appendix B – Standard Field Schema

The table below provides a description of the standard field schema for use in understanding field types and size limits.

Standard Fields	Description (size)
eTrigueID	eTrigue unique ID
EmailAddress	Email used for notifications and alerts (250)
LastName	(100)
FirstName	(100)
Salutation	(20)
Title	(250)
CompanyName	(250)
Address	(250)
Address2	(250)
Address3	(250)
City	(100)
State	(100)
ZipCode	(20)
Country	(100)
Phone	(100)
Phone2	(100)
Fax	(100)
URL	(250)
Memo	Large text
DateAdded	Date time

LastModifiedDate	Date time
Stage__ss	Single Select (Account customizable)
Value__ctyp	ISO currency type (3)
Value__cval	Actual value (float)
crmOwner	From SalesForce (50)
Industry	Free form text (100)
Segment	Free form text (100)
Territory	Free form text (100)
AccountType	Free form text (130)
ExternalID	CRM ID if integrated with CRM; for example SalesForce ID (100)
Source (Profile)	(50)
ProspectOwner	Username of the Prospect Owner
Gender	Single Select (Male or Female)
Status	Single Select (Active, Inactive, Bad Email, or Opt-Out)