LeadSuccess API

The LeadSuccess API provides access to server-side data structures and procedures via a RESTful API to enable 3rd party developers building customized frontend apps to the LeadSuccess System with standard web techniques and is covering the same functionality as it is offered in the original LeadSuccess App.

You can use your 3rd party frontend in combination with original LeadSuccess Apps on mobile devices, mobile scanners of LeadSuccess Service and PC device based LeadSuccess network clients with business card, barcode or form scanners, automatic printer stations, LeadSuccess Kiosk devices or info desk terminals.

The back-office functionality for remote re-editing covers all input devices including 3rd party frontends connected via LeadSuccess API. All reporting and data export variants offered in the LeadSuccess Exhibitor Portal include data collected from all LeadSuccess input devices including 3rd party frontends connected via LeadSuccess API.

You can find more information on http://www.convey.de/ to learn about further components of the LeadSuccess system.

LeadSucces App

Introduction

The LeadSuccess App allows you to collect data and information about the visitors in an easy, fast and reliable way. It allows you to get the visitor´s data in various ways; in fact, it is possible to scan barcodes, QR-Codes, business card or enter information manually. In addition to this, it is also possible to fill out a customizable questionnaire, take notes and add pictures and sketches to every contact you collect.

Exhibitor Portal

Everything can be managed through the “Exhibitor Portal” which allows you to manage your App users, create your own event-specific questionnaire, check and edit leads and furthermore export your collected leads to different Excel formats.

You will need a valid LeadSuccess admin account to configure the event and app users. App user credentials configured in the “Exhibitor Portal” can be used either with the original LeadSuccess App or for requests with basic authentication to use the API.

You will find additional information about administration of LeadSuccess in the document “User Guide LeadSuccess Mobile – Online Portal”

LeadSuccess app structure

The LeadSuccess App user interface offers the following functional areas to the user:

- **Start Page:**
  Event name, user name, number of contacts, present state, new message flag
- **Capture Barcode**
  Create a new contact based on the visitor’s badge barcode
- **Capture Business card**
  Create a new contact based on the visitor’s business card photo and save the photo linked to the contact
- **Edit contact manually**
  Create a new contact or edit existing address fields of a selected visitor contact
- **Edit questionnaire**
  Offers a form to answer the questions of the questionnaire based on the questionnaire configuration (multiselect, single select, combo, rating, text notes, date picker, optional questions, mandatory questions). Each questionnaire is linked to a selected contact
- **Create notes**
  Create and edit sketches in SVG format, capture photos, capture audio messages. All notes are linked to a selected contact
- **Edit user state**
  Edit personal user data, toggle the present state, receive and send user messages and capture a user photo. The data is linked to the employee data of the app user and can be administrated in the Exhibitor Portal

Access protocol

To access this interface is used a subset of ODATA protocol. See ODATA org. ODATA server is based on SAP SQL Anywhere 17 OData implementation. Therefore currently only ODATA version 2.0 syntax can be used. See http://www.odata.org/documentation/odata-version-2-0

LeadSuccess API users will get the **<server-name>** and **<api-name>** to be used for your API-requests together with their login information from convey.

To access this API, HTTPS basic authentication is used. It means user and password has to be transmitted via HTTPS with each request. Since HTTP(S) is a stateless protocol, each request is computed in individual transactions. GET and POST requests return table data as result of the request in the specified format, e.g. JSON. If you use a XMLHTTPRequest alike JavaScript API, use JSON.parse(response.responseText) to transform the result in a JSON object that for automatic data-binding to UI elements within your preferred framework. Some requests will result in URLs to get further data. Due to technical limitations the provided address of these links is a local address and not your API address. Please replace that local address in these URLs to:

```
https://<server-name>/<api-name>/...
```

All parameter data must be transferred in JSON format to the ODATA server. For results the JSON format can be selected with the "$format=json" request option or with the HTTP header “Accept: application/json”, otherwise the server returns the results in XML format. All text data is UTF-8 encoded. All date-time data is in UTC time zone. Pay attention to the fact, that the OData request syntax is **case sensitive**!

A user can only keep one transaction active at a time. If several parallel transactions are to be supported, different users must be used for this!

The following request types are supported

Select data

Use requests of type: **GET** to select data from LeadSuccess relations. You should distinguish at least these different types of GET requests:

Select a list of rows

You can select lists of data from API objects with requests like:

GET https://<server-name>/<api-name>/LSA_<table-name>?$format=json

You can use the $filter=(<filter-list>) and $orderby=(<orderby-list>) options to specify restrictions and list order to the request.

Requests of this kind will return an array of rows, that could be handled like this:

```
function xhrSuccess(response) {
  var obj = JSON.parse(response.responseText);
  var results = obj && obj.d && obj.d.results;
  handleResults(results);
}
```

The result size maximum to be fetched by one request is limited to 100 rows. To fetch the next row-set, you can use the parameter $skiptoken(<primary-key-value>) in a following GET request to the

same relation. To simplify that, you can use the following member in result JSON structure as described above:

```
obj.d.__next
```

Remember to replace the server path in the URL as described above, e.g.:

```
var getNextUrl = function (json) {
    var url = "";
    if (json && json.d) {
        var next = json.d.__next;
        if (next && typeof next === "string") {
            var viewNamePos = next.lastIndexOf("/");
            if (viewNamePos >= 0) {
                url = "https://<server-name>/<api_name>" +
                      next.substr(viewNamePos);
            }
        }
    }
    return url;
}
```

Fetch next list of rows

Use a nextURL given from earlier select request to fetch the next row-set. The returned result is similar to the one of the first select request, including results array and __next member for following row-set, if end of data isn't reached yet.

Select single row

You can use the primary key of each table to select a single row of data with the following request:

```
GET https://<server-name>/<api-name>/LSA_<table-name>(<primary-key-value>)?$format=json
```

Primary keys are always attributes of an integer type.

Requests of this kind will return one row, that could be handled like this:

```
function xhrSuccess(response) {
  var obj = JSON.parse(response.responseText);
  var result = obj && obj.d;
  handleResults(result);
}
```

Insert new data

Use requests of type: **POST** to insert new data into LeadSuccess relations. You can build the request options like this:

```
var newRecord = {
  Fristname: "John",
  LastName: "Doe",
};
```

Remember to omit the primary key value to avoid integrity constraints on insert, unless you need to insert a dependent PK, like for DOC1-relations.

```
var options = {
  type: "POST",
  user: <user-name>,
  password: <password>,
  url: "https://<server-name>/<api-name>/LSA_<table-name>",
  data: JSON.stringify(newRecord),
  headers: {
    "Accept": "application/json",
    "Content-Type": "application/json"
  }
};
```

Requests of this kind will return one row, that could be handled like this:

```
function xhrSuccess(response) {
  var obj = JSON.parse(response.responseText);
  var result = obj && obj.d;
  handleResults(result);
}
```

Pay attention to the fact, that additional columns will be filled automatically by the server side. You need to remember the primary key value returned in the result to select or update the corresponding record.

Update data

Use requests of type: **PUT** to update a specific data row. You can build the request options like this:

```
var options = {
  type: "PUT",
  user: <user-name>,
  password: <password>,
  url: "https://<server-name>/<api-name>/LSA_<table-name>(<primary-key-value>)",
  data: JSON.stringify(newRecord),
  headers: {
    "Accept": "application/json",
    "Content-Type": "application/json"
  }
};
```

Requests of this kind will return a status result on success. To receive the actual record data after the update you need to send an additional GET request for the updated row.

Delete data

Use requests of type: **DELETE** to delete a specific data row. You can build the request options like this:

```
var options = {
  type: "DELETE",
```

```
  user: <user-name>,
  password: <password>,
  url: "https://<server-name>/<api-name>/LSA_<table-name>(<primary-key-value>)",
};
```

Requests of this kind will only return a status result on success.

Procedure call

Use requests of type: **GET** to call procedures. Parameters need to be placed URL-encoded within the URL.

```
var options = {
  type: "GET",
  user: <user-name>,
  password: <password>,
  url: "https://<server-name>/<api-name>/<procedure-name>?<parameters>& $format=json",
};
```

You can build the parameters list like this:

```
<paramaters> = <param1>=<value1>&<param2>=<value2>…;
```

CORS requirements

To support cross-site access-control requests, you should specify the `withCredentials` member of the `XMLHttpRequest` to `true`. You may need to add an option e.g. like this to add the value all your xhr requests, if supported by the `XMLHttpRequest` object:

```
var options = {
  //
  // other options depending from action, see above
  //
  //
  customRequestInitializer: function(req) {
    if (typeof req.withCredentials !== "undefined") {
      req.withCredentials = true;
    }
  }
};
```

Authorization header

To support server-side authorization propagation by some browser clients, like Google Chrome, you should add an “Authorization” header:

```
var options = {
  //
  // other options depending from action, see above
  //
```

```
  //
  headers: {
    "Authorization": "Basic " + btoa(<user> + ":" + <password>)
  }
};
```

Resources

Data categories

Based on the given function areas, the frontend needs to handle different data structures to offer the functionality of the LeadSuccess App. This data can be divided in:

- **Static data**
  to be loaded only once after app installation
- **Event data:**
  to be loaded at least once after app installation and updated after administrative changes in the LeadSuccess Exhibitor Portal
- **Application runtime data**
  to be loaded at least once after app installation and updated after interactive changes in the app or in the LeadSuccess Exhibitor Portal
- **Visitor data**
  to be created interactively in the app and to be retrieved via search and list panes

All data is organized in a relational data structure of tables with primary keys and foreign keys. Each table has a primary key of integer data type named <table-name>ID, that can be referenced as foreign key attribute in another table. The LeadSuccess API offers views called LSA_<table-name> to access the data.

Special relation types

Some naming conventions reflect similar structure and special meaning of specific relation types

- **LGNTINIT<name>**
  used to store translated value lists usually to be used in select elements of the user interface. You need to select the needed translation by specifying LanguageSpecID based on the UI language selection of the user.

- **DOC1<base-table-name>**
  used to store pixel image documents, usually in JPEG format, including up to two optional copies in lower resolution for preview and thumbnail pictures. The document is linked 1:1 to the table <base-table-name> via PK-FK-constraint.

Access rights

The LeadSuccess API offers access rights on object-level to the provided views in order to their usage. For static application and runtime data only GET requests are offered. For writable application runtime data and visitor data POST, PUT and DELETE requests are supported as described to each object. Accessing an object with an unsupported request option will return an exception.

The LeadSuccess API offers access rights on row-level to the provided views. You can only access rows of data in the user context of the currently logged-in user.

Data schema

You can retrieve the full schema data with the following request:

```
GET https://<server-name>/<api-name>/$metadata
```

Static master data

LSA_LanguageSpecExt

Select all entries from this view to get the list of supported languages

Name	Type	Description
LanguageSpecExtID	Int32	Primary key value, use one of these values to select the translation in all relations that need LanguageSpecID to be specified
Title	String(255)	Title of the language
ProjectLanguageID	Int32	Null
ProjectStatus	Int32	0
PrimaryID	Int32	Null
SortID	Int32	Null
Reserved	Int32	Null
DefISOCharset	String(64)	Null
DOMCode	String(31)	Language string to be matched with user selection in web browser
DefaultLanguage	Int32	1|null
ShowLanguage	String(31)	Informational description

Static application runtime data - Select element option lists

LSA_LGNTINITLanguage

Select the entries of this view with filter restriction $filter=(LanguageSpecID eq <primary key value>) to receive a language specific selection for a language select user control.

Name	Type	Description
LGNTINITLanguageID	Int32	Primary key value, no further use
LanguageSpecID	Int32	Foreign key to **LanguageSpecExt**, representing the translation language of the entry
TranslateStatus	Int32	0
INITLangID	Int32	To be used as foreign key in relations referencing LangID
LangTitle	String(32)	Display text for the language to select
LanguageID	Int32	Foreign key to **LanguageSpecExt**, representing the language selection to be done by this entry

LSA_LGNTINITCountry

Select the entries of this view with filter restriction $filter=(LanguageSpecID eq <primary key value>) to receive a language specific selection for a country select user control.

Name	Type	Description
LGNTINITCountryID	Int32	Primary key value, no further use
LanguageSpecID	Int32	Foreign key to **LanguageSpecExt**, representing the translation language of the entry
TranslateStatus	Int32	0
INITCountryID	Int32	To be used as foreign key in relations referencing CountryID
CountryTitle	String(255)	Display text for the country to select
Datfm	String(1)	Internal reference
Ctry1_SAPKey	String(2)	Reference for SAP interface
Ctrya_SAPNumber	String(3)	Reference for SAP interface
Addrs_SAPAddressfunction	String(3)	Reference for SAP interface
Ctryk_VehicleCode	String(3)	Vehicle code
Intca_ISOCode	String(2)	Two letter ISO code
CtryCallingCode	String(50)	Phone number code
ZIPFormat	String(20)	Postal code format identifier
EUCountry	String(1)	Country classification
Imdavers_Number	String(3)	External interface reference
StandardLangID	Int32	Standard language referenced by **LangID**
Natio_Nationalitaet	String(255)	Nationality in text
Domcode	String(31)	Preferred language and country code
Top_Level_Domain	String(5)	Domain code
Alpha3_ISOCode	String(3)	Three letter ISO code
Numeric3_ISOCode	String(3)	Numeric ISO code

You can use several reference columns to identify country selection for contact data import / export.

LSA_LGNTINITAddress

Select the entries of this view with filter restriction $filter=(LanguageSpecID eq <primary key value>) to receive a language specific selection for a form-of select user control.

Name	Type	Description
LGNTINITAddressID	Int32	Primary key value, no further use
LanguageSpecID	Int32	Foreign key to **LanguageSpecExt**, representing the translation language of the entry
TranslateStatus	Int32	0
INITAddressID	Int32	To be used as foreign key in relations referencing **AddressID**
AddressTitle	String(32)	Display text for the form-of to select
LetterAddress	String(50)	Display text for letter address

LSA_LGNTINITUserPresence

Select the entries of this view with filter restriction $filter=(LanguageSpecID eq <primary key value>) to receive a language specific selection for a select user control to offer a choice of employee's next time return to presence.

Name	Type	Description
LGNTINITUserPresenceID	Int32	Primary key value, no further use
LanguageSpecID	Int32	Foreign key to **LanguageSpecExt**, representing the translation language of the entry
TranslateStatus	Int32	0
INITUserPresenceID	Int32	To be used as foreign key in relations referencing **UserPresenceID**
PresenceTitle	String(32)	Display text for the next time return to presence

LSA_LGNTINITOptionType

Select the entries of this view with filter restriction $filter=(LanguageSpecID eq <primary key value>) to receive a language specific selection for available event options.

Name	Type	Description
LGNTINITOptionTypeID	Int32	Primary key value, no further use
LanguageSpecID	Int32	Foreign key to **LanguageSpecExt**, representing the translation language of the entry
TranslateStatus	Int32	0
INITOptionTypeID	Int32	To be used as foreign key in relations referencing **OptionTypeID**
OptionTypeTitle	String(500)	Display text for the event option type

Read only application runtime data

Select this data to show information relevant for application runtime.

LSA_LOADEvent

Select the entry of this view to show information about the current event.

Name	Type	Description
LOADEventID	Int32	Primary key value. Use this value as foreign key reference in relations referencing the event by **EventID**
EventName	String(2147483647)	Display text for the event
Startdate	DateTime	Start date of event
Enddate	DateTime	End date of event
Active	Int32	Internal use
Questionnaire	String(255)	Internal use
EventTitle	String(2147483647)	Additional event text
Subtitle	String(2147483647)	Additional event text
EventText_1	String(2147483647)	Additional event text
EventText_2	String(2147483647)	Additional event text
EventText_3	String(2147483647)	Additional event text

Name	Type	Description
OrganiserID	Int32	Internal use
ExtRef	String(2147483647)	External reference
LayoutNew	Int32	Internal use
DBSYNCLogin	String(5)	Internal use
DBSYNCPassword	String(3)	Internal use
FairMandantEventID	String(3)	Internal use
FairMandantID	Int32	Internal use
CompanyName	String(2147483647)	Address data of exhibitor
Street	String(2147483647)	Address data of exhibitor
ZIP	String(2147483647)	Address data of exhibitor
City	String(2147483647)	Address data of exhibitor
INITCountryID	Int32	Address data of exhibitor
Website	String(2147483647)	Address data of exhibitor
PrivacyPolicyText	String(2147483647)	Privacy policy text of exhibitor
PrivacyPolicySVG	String(2147483647)	Privacy policy text of exhibitor used for standard note in SVG format

LSA_CREventOption

Select the entries of this view to show information about the options available for the current event.

Name	Type	Description
CREventOptionID	Int32	Primary key value, no further use
EventID	Int32	Foreign key to **Event**, representing the current event
INITOptionTypeID	Int32	Foreign key to **OptionType,** representing the current event option type
LocalValue	String(500)	String value of the option
NumberValue	Int32	Number value of the option

LSA_Employee

Select the entry of this view to show user related information useful for a start page of the app.

Name	Type	Description
EmployeeID	Int32	Primary key value, use this id as foreign key for user related references
LoginName	String(255)	Login name of the user
EmployeeName	String(2147483647)	Display name of the user
EventID	Int32	Foreign key to **Event,** representing the current event
LocalContactsCount	Int16	0
SentContactCount	Int32	Number of contacts related to current user that have been send to the LeadSuccess server
Present	Int32	1 if present on site
TimeZoneAdjustment	Int16	0
CurrentTS	DateTime	Current timestamp of data
NewNoteCount	Int16	>0 if new message from info desk is available

LSA_MandatoryFields

Select the entries of this view to get a list of mandatory address fields

Name	Type	Description
MandatoryFieldsID	Int32	Primary key value, no further use
MandatoryFieldTypeID	Int32	Internal use
EventID	Int32	Foreign key to **Event,** representing the current event
FieldFlag	Int32	1 if the field is mandatory
AttributeName	String(31)	Name of the attribute of the current contact record that is to be handled as mandatory

LSA_LineAnswerData

Select the entries of this view to get a list of questions and answer options to be used to show the questionnaire related to the current contact. Use the following restriction:

`$filter=(ContactID eq <primary key of current contact>)&$orderby=(SortIndex)`

Name	Type	Description
LineAnswerDataID	Int32	Primary key value. Use this value as primary key reference to update the question row
QuestionnaireLineID	Int32	Internal use
ContactID	Int32	Foreign key **ContactID,** representing the current contact
SSAnswer	String(2)	Selected single selection answer
MSAnswer01…28	String(1)	Selected multi selection answers, value “X” signals if the answer is selected
FreeText	String(1000)	Optional text comment
MRAnswer01…06	String(1)	Selected multi rating answers, value “X” signals if the answer is selected
RRAnswer	String(2)	Selected single rating answer
Question	String(5005)	Question text
SortIndex	Int32	Sort order of questions
FreeTextActive	String(1)	1 if question text field has to be shown
DividingLineFlag	String(1)	1 if a dividing line has to be shown
QuestionGroup	String(1023)	Question group title text
MS01…28	String(100)	Answer display text for multi selection answers
SSAnswer01…28	String(2)	Available answer value for single selection answers
SS01…28	String(100)	Answer display text for single selection answers
SR01…06	String(32)	Answer display text for single rating answers
MR01…06	String(32)	Answer display text for multi rating answers
SRMax	Int32	Number of available single rating answers
MRShow01…06	Int32	1 if corresponding multi rating answer has to be shown
Combobox	String(1)	1 if a combobox control has to be shown for single selection
DateCombobox	String(1)	1 if a date picker control has to be shown for additional answer text

Name	Type	Description
DOC1LineAnswerID	Int32	Reference to question related image data
MandatoryField	Int32	1 if the question is mandatory
SelectedQuestionIdx	Int32	SortIdx of the question that enables the current question. If not enabled the current question has to be hidden.
SelectedAnswerIdx	Int32	Index of the answer of the question that enables the current question. If not enabled the current question has to be hidden.

Writeable application runtime data

LSA_UserInfo

Select and/or update the entry of this view to retrieve or change user info and messages

Name	Type	Description
UserInfoID	Int32	Primary key value, 1:1 reference to **EmployeeID**
TagID	String(255)	Internal use
Lastname	String(255)	Last name of the user
Present	Int32	1 if the user is present on site
PosX	Decimal	Internal use
PosY	Decimal	Internal use
Info1	String(255)	User message text
Info2	String(255)	Info desk message text
INITUserPresenceID	Int32	Foreign key reference **UserPresenceID**
Info1TS	DateTime	Edit timestamp of user message
Info2TS	DateTime	Edit timestamp of info desk message
Info1TSRead	DateTime	Read timestamp of user message
Info2TSRead	DateTime	Read timestamp of info desk message
Firstname	String(64)	First name of the user
UserTitle	String(80)	Title of the user
Position	String(80)	Position of the user
Telefon	String(64)	Phone number of the user
TelefonMobile	String(64)	Mobile phone number of the user
Email	String(500)	Email address of the user

LSA_DOC1Employee

Select and/or insert/update/delete the entry of this view to retrieve or change a user image

Name	Type	Description
DOC1EmployeeID	Int32	Primary key value, 1:1 reference to **EmployeeID**
wFormat	Int32	3
ColorType	Int32	11
ulWidth	Int32	Width of image in pixel
ulHeight	Int32	Height of image in pixel
ulDpm	Int32	Resolution of image in dots per meter
szOriFileNameDOC1	String(255)	Original filename
szDocPathDOC2	String(255)	Internal use

Name	Type	Description
szOvwPathDOC3	String(255)	Internal use
szPrevPathDOC4	String(255)	Internal use
ulOvwEdge	Int32	Maximum edge size of thumbnail
ulPrevEdge	Int32	Maximum edge size of preview
DocContentDOCCNT1	String(2147483647)	Base64 encoded image data
PrevContentDOCCNT2	String(2147483647)	Base64 encoded image data of preview
OvwContentDOCCNT3	String(2147483647)	Base64 encoded image data of thumbnail
ContentEncoding	Int32	4096

Visitor data

LSA_Contact

Select and/or insert/update/delete the entry of this view to retrieve or change visitor contact data. You need to insert an empty contact first before inserting barcode or business card data related to that contact.

Name	Type	Description
ContactID	Int32	Primary key value, 1:1 reference to **EmployeeID**
ContactTitle	String(80)	Title
FirstName	String(256)	First name
LastName	String(128)	Last name
Position	String(80)	Position at work
CompanyName	String(1024)	Company name
Street	String(128)	Street address
POBox	String(32)	Post office box number
ZIP	String(12)	Postal code of address
City	String(128)	Name of city
State	String(128)	Name of federal state or country
INITCountryID	Int32	Foreign key reference **CountryID** for country selection
Phone	String(64)	Phone number
Mobile	String(64)	Mobile phone number
Fax	String(64)	Facsimile number
Email	String(500)	Email address
Website	String(500)	Web site address
EmployeeID	Int32	Foreign key reference **EmployeeID** specifying the owner of the contact
CreateDate	DateTime	Creation timestamp
Comment	String(4000)	Comment
ActionID	Int32	Internal use
StatusID	Int32	Internal use
INITAddressID	Int32	Foreign key reference **AddressID** for form-of selection
Incomplete	Int32	1 if not yet being reviewed in back-office
EventID	Int32	Foreign key reference **EventID** specifying the reference to the current event
Middlename	String(64)	Middle name
ImportCardscanID	Int32	Foreign key reference **ImportCardscanID** specifying the reference to a business card

Name	Type	Description
HostName	String(255)	Unique identifier of your device. Internal use to distinguish your devices
POBoxZIP	String(12)	Postal code of post office box number
Freetext1…6	String(255)	Additional comments
ScanFlag	Int32	Internal use
INITDepartmentID	Int32	Internal use
QuestionnaireBatchID	Int32	Internal use
DepartmentText	String(4000)	Workplace department
Sector	String(4000)	Workplace sector
MasterContactID	Int32	Internal use

LSA_ImportBarcodeScan

Insert a new entry, if you want to recognize contact data from one dimensional barcode data.

Name	Type	Description
ImportBarcodeScanID	Int32	Primary key value, no further use
Request_Barcode	String(32)	Barcode data
ContactID	Int32	Foreign key **ContactID,** representing the current contact

LSA_ImportCardscan

Insert a new entry, if you want to recognize contact data from business card image or scanned two dimensional code data.

Name	Type	Description
ImportCardscanID	Int32	Primary key value. Use this value as primary key reference for Insert of **DOC1ImportCardscan** in case of business card image
Button	String(100)	“OCR_TODO” in case of business card image or “VCARD_TODO” in case of two dimensional code
Barcode2	String(2147483647)	Data content of two dimensional code
ContactID	Int32	Foreign key **ContactID,** representing the current contact

LSA_DOC1ImportCardscan

Insert a new entry containing base64 encoded image data, if you want to recognize contact data from business card image.

Name	Type	Description
DOC1ImportCardscanID	Int32	Primary key value, 1:1 reference to **ImportCardscanID**
wFormat	Int32	3
ColorType	Int32	11
ulWidth	Int32	Width of image in pixel
ulHeight	Int32	Height of image in pixel
ulDpm	Int32	Resolution of image in dots per meter

Name	Type	Description
szOriFileNameDOC1	String(255)	Original filename
szDocPathDOC2	String(255)	Internal use
szOvwPathDOC3	String(255)	Internal use
szPrevPathDOC4	String(255)	Internal use
ulOvwEdge	Int32	Maximum edge size of thumbnail
ulPrevEdge	Int32	Maximum edge size of preview
DocContentDOCCNT1	String(2147483647)	Base64 encoded image data
PrevContentDOCCNT2	String(2147483647)	Base64 encoded image data of preview
OvwContentDOCCNT3	String(2147483647)	Base64 encoded image data of thumbnail
ContentEncoding	Int32	4096

PRC_GetRecognizedContact

Call this procedure to get updated contact data extracted from previously inserted barcode or business card image. You need to INSERT the data into **LSA_Contact** and **LSA_ImportBarcodeScan** or **LSA_Contact** and **LSA_ImportCardscan** or **LSA_Contact** and **LSA_ImportCardscan** and **LSA_DOC1ImportCardscan** first. The procedure will wait for data computation up to the given number of seconds.

Parameter	Type	Description
p_ContactID	Int32	Key value **ContactID,** representing the current contact
p_TimeoutSec	Int32	Timeout in seconds, maximum value 59s

If the contact is processed within the timeout, the procedure will return the contact data, like a single row select. If no data will be returned within the timeout, you need to select the contact data later again via single or multiple row select from **LSA_Contact**.

The procedure described here allows to enter additional data to your contact before the recognized data from barcode or business card analysis is returned. Your app users could edit answers to the lead questionnaire or enter notes, sketches, photos or audio recordings in addition to the contact data without a delay. Your app could call the procedure **PRC_GetRecognizedContact** in the background to complete your contact address information after the server has computed the contact data retrieval request which may take some time. This enables fluent usage of the API functionality without a break in the flow of user operation to wait for incoming data for several seconds.

PRC_GetContactFromBarcode

Call this procedure to get a new contact data record extracted from 1d-barcode or extracted barcode ID from 2d/QR-code text, e.g. in VCARD field X-REFCODE. This procedure is a shortcut version of the procedure described above. Using this procedure, you don't need to insert data into **LSA_Contact** and **LSA_ImportCardscan** first. As a tradeoff you cannot add additional data to the contact data before this procedure returned with the newly inserted **ContactID**. The procedure will wait for data computation up to the given number of seconds.

Parameter	Type	Description
p_EmployeeID	Int32	Foreign key reference **EmployeeID** specifying the owner of the contact. You need to select this value from **LSA_Employee** first
p_Request_Barcode	String(1000)	Barcode data or extracted 2d/QR-code text from field X-REFCODE
p_TimeoutSec	Int32	Timeout in seconds, maximum value 59s
p_EventID	Int32	Foreign key reference **EventID** specifying the reference to the current event. You need to select this value from **LSA_Event** first
p_HostName	String(255)	Unique identifier of your device. Internal use to distinguish your devices
p_Incomplete	Int32	1 if not yet being reviewed in back-office

If the contact is processed within the timeout, the procedure will return the contact data, like a single row select. If no data will be returned within the timeout, you need to select all newly inserted contact data records later again via multiple row select from **LSA_Contact**.

Questionnaire

LSA_LineAnswer

Use this view to update row data of one question retrieved previously from a select from **LSA_LineAnswerData**. You need to insert a contact first before you can select the related questionnaire and update the rows in **LSA_LineAnswer**.

Name	Type	Description
LineAnswerID	Int32	Primary key value. Use this value as primary key reference to update the question row, 1:1 reference to **LineAnswerDataID**
QuestionnaireLineID	Int32	Internal use
ContactID	Int32	Foreign key **ContactID,** representing the current contact
SSAnswer	String(2)	Selected single selection answer
MSAnswer01…28	String(1)	Selected multi selection answers, value "X" signals if the answer is selected
FreeText	String(1000)	Optional text comment
MRAnswer01…06	String(1)	Selected multi rating answers, value "X" signals if the answer is selected
RRAnswer	String(2)	Selected single rating answer

LSA_DOC1LineAnswer

Insert entries to add image data related to the questionnaire

Name	Type	Description
DOC1LineAnswerID	Int32	Primary key value, 1:1 reference to **LineAnswerID**
wFormat	Int32	3
ColorType	Int32	11
ulWidth	Int32	Width of image in pixel
ulHeight	Int32	Height of image in pixel
ulDpm	Int32	Resolution of image in dots per meter
szOriFileNameDOC1	String(255)	Original filename
szDocPathDOC2	String(255)	Internal use
szOvwPathDOC3	String(255)	Internal use
szPrevPathDOC4	String(255)	Internal use
ulOvwEdge	Int32	Maximum edge size of thumbnail
ulPrevEdge	Int32	Maximum edge size of preview
DocContentDOCCNT1	String(2147483647)	Base64 encoded image data
PrevContentDOCCNT2	String(2147483647)	Base64 encoded image data of preview
OvwContentDOCCNT3	String(2147483647)	Base64 encoded image data of thumbnail
ContentEncoding	Int32	4096

LSA_ContactNote

Select and/or insert/update/delete the entry of this view to retrieve or change notes related to visitor contact data. You need to insert a contact first before inserting related notes.

Name	Type	Description
ContactNoteID	Int32	Primary key value, used to reference the note
NoteTitel	String(255)	Optional title
DateCreated	DateTime	Creation timestamp
ContactID	Int32	Foreign key **ContactID,** representing the current contact
ExecAppTypeID	Int32	3 – in case of image data 15 – in case of SVG file data with graphical note 16 – in case of audio data
SourceCode	String(2147483647)	Document source: Base64 encoded image data in case if image document, Base64 encoded audio data file in case of audio note, SVG text source in case of graphical sketch note
DocGroup	Int32	1 – in case of image data 3 – in case of SVG file data with graphical note 6 – in case of audio data
DocFormat	Int32	Format retrieved from file name extension or content-type
OvwSourceCode	String(2147483647)	Optional base 64 encoded thumbnail image
PrvSourceCode	String(2147483647)	Optional base 64 encoded preview image
Width	Int32	Width of image in pixel
Height	Int32	Height of image in pixel

Name	Type	Description
Length	Int32	Length of audio message in ms
OvwEdge	Int32	Maximum edge size of thumbnail
DocExt	Int32	Document filename extension
ColorType	Int32	11 – in case of image data

You can use the following look-ups to retrieve the DocFormat attribute:

```
docFormat: {
"bmp": 1,
"tif": 2,
"tiff": 2,
"jpg": 3,
"jpeg": 3,
"pcx": 4,
"gif": 5,
"eps": 6,
"ps": 7,
"dcx": 10,
"avi": 13,
"wav": 14,
"pcd": 15,
"wmf": 16,
"cdr": 17,
"plt": 18,
"dxf": 21,
"dwg": 22,
"hgl": 23,
"hpgl": 23,
"cgm": 24,
"ctm": 25,
"ch3": 26,
"cgt": 27,
"clp": 28,
"clx": 29,
"rtf": 30,
"htm": 31,
"html": 31,
"wri": 32,
"drw": 33,
"dsf": 34,
"dwf": 35,
"flw": 36,
"fmv": 37,
"fpx": 38,
"gal": 39,
"gdf": 40,
"gem": 41,
"hgw": 42,
"ico": 43,
"igs": 44,
"img": 45,
"met": 46,
"mnp": 47,
"nap": 48,
"pct": 49,
"pdf": 50,
"pic": 51,
"pif": 52,
"png": 53,
"ppt": 54,
"psd": 55,
"ras": 56,
"rnd": 57,
"sat": 58,
"shw": 59,
"mid": 60,
"midi": 60,
"tga": 61,
"vsd": 62,
"wpg": 63,
"xbm": 64,
"xpm": 65,
"xwd": 66,
"mp3": 67,
"mpa": 67,
"m2a": 67,
"m3a": 67,
"m4a": 67,
"3g2": 67,
"3gp": 67,
"3ggp": 67,
"3gpp": 67,
"mov": 68,
"au": 69,
"asc": 70,
"ascii": 70,
"txt": 70,
"aif": 71,
"aiff": 71,
"mpg": 72,
"mpeg": 72,
"cmx": 73,
"can": 74,
"canv": 74,
"svg": 75,
"svgz": 76,
"svz": 76,
"emf": 77,
"asf": 81,
"wma": 82,
"wmv": 83,
"swf": 1704
},

docFormatContentType: {
1: "image/bmp",
2: "image/tiff",
3: "image/jpeg",
5: "image/gif",
6: "application/postscript",
7: "application/postscript",
13: "video/avi",
14: "audio/wav",
16: "image/x-wmf",
30: "application/msword",
31: "text/html",
50: "application/pdf",
53: "image/png",
67: "audio/mpeg",
68: "video/quicktime",
72: "video/mpeg",
82: "audio/x-ms-wma",
83: "video/x-ms-wmv"
}
```

LSA_ContactReport

Select all entries from this view to get the list of visitor contact data with related questionnaire.

Name	Type	Description
ContactReportID	Int32	Primary key value -2 = line with question text and field headings -1 = line with group names >0 = line with contact data
ContactID	String(32)	Foreign key **ContactID,** representing the current contact
Address	String(32)	Address
Title	String(80)	Title
FirstName	String(256)	First name
MidleName	String(256)	Middle name
LastName	String(128)	Last name
Position	String(80)	Position at work
Division	String(4000)	Company division
CompanyName	String(1024)	Company name
Sector	String(4000)	Sector
Street	String(128)	Street address
POBox	String(32)	Post office box number
POBox_Zip	String(12)	Post office box number with postal code
ZIP	String(12)	Postal code of address
City	String(128)	Name of city
State	String(128)	Name of federal state or country
Country	String(255)	Country
Phone	String(64)	Phone number
Mobile	String(64)	Mobile phone number
Fax	String(64)	Facsimile number
Email	String(500)	Email address
Website	String(500)	Web site address
Memo	String(4000)	Memo
RecordedBy	String(194)	Employee name - specifying the owner of the contact
RecordDate	String(32)	Creation timestamp
Remarks	String(4000)	Comment
EntryComplete	String(32)	1 if not yet being reviewed in back-office
Event	String(127)	Name of the current event
Question01	String(2000)	Answer to Question 1
...	..	.. to
Question60	String(2000)	Answer to Question 60
RequestBarcode	String(32)	Request barcode
ModificationDate	String(32)	Modification date
QuestionnairScans	String(32)	Count of questionnaire scans
Sketches	String(32)	Count of sketches

LSA_ContactPdfExport

Select entries to get Pdf data of contact questionnaire

Name	Type	Description
ContactPdfExportID	Int32	Primary key value
wFormat	Int32	3
szOriFileNameDOC1	String(255)	Original filename
szOriPrevFileNameDOC2	String(255)	Internal use
szOriOverFileNameDOC3	String(255)	Internal use
szDocPathDOC4	String(255)	Internal use
szOvwPathDOC5	String(255)	Internal use
szPrevPathDOC6	String(255)	Internal use
ulOvwEdge	Int32	Maximum edge size of thumbnail
ulPrevEdge	Int32	Maximum edge size of preview
DocContentDOCCNT1	String(2147483647)	Base64 encoded image data
PrevContentDOCCNT2	String(2147483647)	Base64 encoded image data of preview
OvwContentDOCCNT3	String(2147483647)	Base64 encoded image data of thumbnail
ContentEncoding	Int32	4096
EntryComplete	String(32)	1 if not yet being reviewed in back-office
RecordedBy	String(194)	Employee name - specifying the owner of the contact
RecordDate	String(32)	Creation timestamp
ContactID	Int32	Primary key value, 1:1 reference to **ContactID**

Error codes

For all requests, a result code and possibly a textual error message is returned.

The error message serves to describe the error for a developer as exactly as possible. Ideally, it should be saved in case of error in a log file. It is not intended to be displayed to an end user (exhibitor).

The result code is roughly based on the HTTP status codes and is intended as the basis for an automatic response to specific errors. The three-digit numeric error code can be followed by a detail error code with a dot.

The rough classification of the error is made possible by the first digit of the result code:

• "2" for success

• "4" for client-side errors, e.g. wrong or missing parameters

• "5" for server-side errors

Common error codes used by the API:

• "400" Bad Request: Incorrect request, e.g. missing mandatory parameter

• "404" Not Found: No matching record was found. If appropriate, the type of missing data record is specified as the detail error code:

• "500" Internal Server Error: There is a problem on the server, e.g. missing documents. Here a LeadSuccess administrator should be contacted. The plain text message helps to localize the problem!

Please note that in addition to the application error codes defined here, further error messages of the overlying protocol levels (ODATA, web server) can occur.

Version Overview:

Version Manual	Version Database	Date	Changes
0.1	7.0.15	2018-09-19	1st pre-release of API manual
0.2	7.0.15	2018-09-25	New: Procedure call, CORS, PRC_GetRecognizedContact Changed: column LSA_Contact.FirstName
0.3	7.0.15	2018-09-26	New: Authorization header
0.4	7.0.15	2018-10-02	Changed: Next-URL
0.5	7.0.15	2018-10-05	Changed: $filter using “eq” instead of “=”
0.6	7.0.19	2019-01-25	New View for Contact Report and PDF
0.7	No change	2022-09-19	Changed typo: PRC_GetRecognizedContact (case-sensitive)
0.8	No change	2024-03-12	Changed wrong data types, typos in **LSA_Contact**
0.9	8.5.01	2024-06-25	Added **PRC_GetContactFromBarcode**
			.