AvailableFields class

- Full class name: `GS\SmartSearch\AvailableFields`

Information:

- Type: Constant Definitions

This class represents available fields for the smart search.

Available Constants

- `NAME_OR_PHONE` - string, Represents the 'Phone Or Company Name' smart search field.
- `TAGS` - string, Represents the 'Tags' smart search field.
- `CONTACT_LIST` - string, Represents the 'Contact List' smart search field.
- `ORGANIZATION` - string, Represents the 'Company' smart search field.
- `TITLE` - string, Represents the 'Title' smart search field.
- `CITY` - string, Represents the 'City' smart search field.
- `STATE` - string, Represents the 'State' smart search field.
- `COUNTRY` - string, Represents the 'Country' smart search field.
- `LOCATION` - string, Represents the 'Location' smart search field.
- `SOCIAL_CONTENT` - string, Represents the 'Social Content' smart search field.
- `EMPLOYEE` - string, Represents the 'Assigned User' smart search field.
- `SUBSCRIBE_STATUS` - string, Represents the 'Subscribe Status' smart search field.
- `ADDED_TIME` - string, Represents the 'Added Time' smart search field.
- `REGISTER_AFTER` - string, Represents the 'Registered After' smart search field.
- `REGISTER_BEFORE` - string, Represents the 'Registered Before' smart search field.
- `TOUCHES` - string, Represents the 'Touches' smart search field.
- `CONFIDENCE_LEVEL` - string, Represents the 'Confidence Level' smart search field.

`Fields(): array`

Returns an array of available smart search fields.

Parameters: None

Returns:

- `array` - An array containing available smart search fields.

`FieldsValues(): array`

Returns an array of field values corresponding to smart search fields.

Parameters: None

Returns:

- `array` - An array containing field values corresponding to smart search fields.

`Titles(): array`

Returns an array of titles for smart search fields.

Parameters: None

Returns:

- `array` - An array containing titles for smart search fields.

`GetFieldTagByFieldValue(string $fieldValue)`

Returns the field tag based on the provided field value.

Parameters:

- `string $fieldValue` - The field value for which the corresponding field tag is needed.

Returns:

- The field tag associated with the provided field value (one of the described constants).

`GetFieldValue(string $tagname, $value)`

Returns the field value formatted for the search form based on the provided field tag (one of the constants) and value.

Parameters:

- `string $tagname` - The field tag.
- `$value` - The field value.

Returns:

- The field value.

`FieldsForTag(string $tagname)`

Returns an array of field configurations based on the provided field tag (one of the constants). The field configurations can then be used for building an HTML form.

Parameters:

- `string $tagname` - The field tag.

Returns:

- An array containing field configuration based on the provided field tag.

Company class

The `Company` class represents a business that has a subscription to PushButton CRM. Every object registered in CRM (users, employees, leads, customers, etc.) and every action that takes place within CRM is linked to the `Company`.

- Full class name: `\GS\Company\Company`
- Legacy alias (shouldn't be used in a new code): ~~`\TCompany`~~

Information:

- Type: ORM Model
- DB Table: `companies`

`\GS\Company\Company` has the responsibility for manipulating a single company record in a database. It contains additional methods for managin company properties.

Table of Contents

Common

`ID()`

Gets the company ID.

Returns: Company ID.

`Status()`

Gets the company status.

Returns: Company status.

`Exists()`

Checks if the company exists.

Returns: `true` if the company exists, otherwise `false`.

`Name()`

Gets the name of the company.

Returns: Company name.

`SetName($val)`

Sets the name for the company.

Parameters:

- `$val` - Name to set for the company.

`Initials()`

Returns the initial of the company name (i.e. **A** for **Abcd Inc**).

Returns: String

`UpdateValues($params)`

Updates multiple values for the current customer.

Parameters:

- `$params` - An associative array of key-value pairs representing the values to be updated.

`UsingCache($id)`

Returns a cached instance of the `Company` class for a given ID.

Parameters:

- `$id` - Integer ID of the company.

Returns: TCompany

`Remove()`

Deletes the company and all assigned resources.

Shortcuts

`CurrentCompany()`

Returns the currently set company as a singleton object.

`SystemCompany()`

Returns the system company as a singleton object.

Initialization

`InitCurrentCompany($id)`

Initializes and returns the current company based on the provided ID.

Parameters:

- `$id` - ID of the company.

`initWithEmail($email)`

Initializes and returns a company based on the email.

Parameters:

- `$email` - Email address.

`initWithSubscriptionID($tsubscription_id)`

Initializes and returns a company based on the subscription ID.

Parameters:

- `$tsubscription_id` - subscription ID

`initWithPhone($twiliophone)`

Initializes and returns a company based on the Twilio phone number.

Parameters:

- `$twiliophone` - Twilio phone number.

`checkIfExistWithPhone($twiliophone)`

Checks if a company with the given Twilio phone number exists and returns the corresponding company object.

Parameters:

- `$twiliophone` - Twilio phone number to check.

Returns: Company object if exists, otherwise `false`.

`initWithKnowledgebaseDomain($domain)`

Initializes and returns a company based on the knowledge base domain.

Parameters:

- `$domain` - Knowledge base domain.

`initWithHubSpotID($portalId)`

Initializes and returns a company based on the HubSpot ID.

Parameters:

- `$portalId` - HubSpot portal ID.

`initWithID($company_id)`

Initializes and returns a company based on the company ID.

Parameters:

- `$company_id` - Company ID.

`initWithMobileAppAuthorizationCode($code)`

Initializes and returns a company based on the mobile app authorization code.

Parameters:

- `$code` - Mobile app authorization code.

Returns: Company object if successful, otherwise `false`.

In-App Calls

`isInAppCallsEnabled()`

Checks if in-app calls are enabled for the company.

Returns: `true` if in-app calls are enabled, otherwise `false`.

`SetInAppCallsStatus($bool_val)`

Sets the status of in-app calls for the company.

Parameters:

- `$bool_val` - Boolean value to set for in-app calls status.

Initial Message

`CanSendInitialMessages()`

Checks if the company can send initial messages.

Returns: `true` if initial messages can be sent, otherwise `false`.

`SetInitialMessage($bool_val)`

Sets the status of sending initial messages for the company.

Parameters:

- `$bool_val` - Boolean value to set for sending initial messages.

Limitations

`TextMessagesLeft()`

Calculates and returns the remaining number of text messages that can be sent based on the set limit.

Returns: Remaining number of text messages.

`EmailMessagesLeft()`

Calculates and returns the remaining number of email messages that can be sent based on the set limit.

Returns: Remaining number of email messages.

`VoiceMessagesLeft()`

Calculates and returns the remaining number of voice messages that can be sent based on the set limit.

Returns: Remaining number of voice messages.

`CanSendTextMessages()`

Checks if the company can send text messages based on the configured limits.

Returns: `true` if text messages can be sent, otherwise `false`.

`CanSendVoiceMessages()`

Checks if the company can send voice messages based on the configured limits.

Returns: `true` if voice messages can be sent, otherwise `false`.

`CanSendEmailMessages()`

Checks if the company can send email messages based on the configured limits.

Returns: `true` if email messages can be sent, otherwise `false`.

`HasTextLimit()`

Checks if the company has a text message limit.

Returns: Boolean

`HasEmailLimit()`

Checks if the company has an email limit.

Returns: Boolean

HasVoiceLimit()

Checks if the company has a voice call limit.

Returns: Boolean

TextLimit()

Returns the text message limit of the company.

Returns: Integer

EmailLimit()

Returns the email limit of the company.

Returns: Integer

VoiceLimit()

Returns the voice call limit of the company.

Returns: Integer

TextUsed()

Returns the number of text messages used by the company.

Returns: Integer

EmailUsed()

Returns the number of emails used by the company.

Returns: Integer

VoiceUsed()

Returns the number of voice calls used by the company.

Returns: Integer

SetTextLimit($val)

Updates the text message limit for the current company.

Parameters:

- `$val` - Integer value.

`SetEmailLimit($val)`

Updates the email limit for the current company.

Parameters:

- `$val` - Integer value.

`SetVoiceLimit($val)`

Updates the voice call limit for the current company.

Parameters:

- `$val` - Integer value.

`SetTextUsed($val)`

Updates the number of text messages used for the current company.

Parameters:

- `$val` - Integer value.

`SetEmailUsed($val)`

Updates the number of emails used for the current company.

Parameters:

- `$val` - Integer value.

`SetVoiceUsed($val)`

Updates the number of voice calls used for the current company.

Parameters:

- `$val` - Integer value.

Business Type

`IdBusinessType()`

Gets the ID of the business type associated with the company.

Returns: ID of the business type.

`HasIdBusinessType()`

Checks if the company has a business type.

Returns: `true` if a business type is set, otherwise `false`.

`SetBusinessTypeID(val)`

Sets the business type ID for the company.

Parameters:

- `val` - Value to set.

Returns: None

`CloneBusinessTypeSequences($businesstype)`

Clones business type sequences associated with the company.

Parameters:

- `$businesstype` - Business type object.

Returns: None

`CloneBusinessTypeTemplates($businesstype)`

Clones business type templates associated with the company.

Parameters:

- `$businesstype` - Business type object.

Returns: None

Subscription

`ExpirationTimestamp()`

Gets the expiration timestamp for the company.

Returns: Expiration timestamp.

`isUnsubscribeMessageSet()`

Checks if the unsubscribe message is set for the company.

Returns: `true` if the unsubscribe message is set, otherwise `false`.

`HasSubscriptionID()`

Checks if the company has a subscription ID.

Returns: `true` if a subscription ID is set, otherwise `false`.

`SubscriptionID()`

Gets the subscription ID of the company.

Returns: Subscription ID.

`SetSubscriptionID($id_subscription)`

Sets the subscription ID for the company.

Parameters:

- `$id_subscription` - Subscription ID to set.

`SetExpirationTimestamp($expiration)`

Sets the expiration timestamp for the company.

Parameters:

- `$expiration` - Expiration timestamp to set.

Address

`HasAddress()`

Checks if the company has an address.

Returns: `true` if an address is set, otherwise `false`.

`Address()`

Gets the address of the company.

Returns: Company address.

`SetAddress($val)`

Sets the address for the company.

Parameters:

- `$val` - Address to set for the company.

`HasAddress2()`

Checks if the company has a second address.

Returns: `true` if a second address is set, otherwise `false`.

`Address2()`

Gets the second address of the company.

Returns: Second address.

`SetAddress2($val)`

Sets the second address for the company.

Parameters:

- `$val` - Second address to set.

`HasCity()`

Checks if the company has a city.

Returns: `true` if a city is set, otherwise `false`.

`City()`

Gets the city of the company.

Returns: Company city.

`SetCity($val)`

Sets the city for the company.

Parameters:

- `$val` - City to set for the company.

`HasState()`

Checks if the company has a state.

Returns: `true` if a state is set, otherwise `false`.

`State()`

Gets the state of the company.

Returns: Company state.

`SetState($val)`

Sets the state for the company.

Parameters:

- `$val` - State to set for the company.

`HasZip()`

Checks if the company has a zip code.

Returns: `true` if a zip code is set, otherwise `false`.

`Zip()`

Gets the zip code of the company.

Returns: Company zip code.

`SetZip($val)`

Sets the zip code for the company.

Parameters:

- `$val` - Zip code to set for the company.

HubSpot

`SetHubSpotDefaultCustomerStatus($val)`

Sets the default customer status for HubSpot integration.

Parameters:

- `$val` - Default customer status to set for HubSpot integration.

`HubSpotDefaultCustomerStatus()`

Gets the default customer status for HubSpot integration.

Returns: Default customer status for HubSpot integration.

`isHubSpotConnected()`

Checks if the company is connected to HubSpot.

Returns: `true` if connected to HubSpot, otherwise `false`.

`HasHubSpotAccessToken()`

Checks if the company

has a HubSpot access token.

Returns: `true` if a HubSpot access token is set, otherwise `false`.

`HubSpotAccessToken()`

Gets the HubSpot access token for the company.

Returns: HubSpot access token.

`SetHubSpotAccessToken($val)`

Sets the HubSpot access token for the company.

Parameters:

- `$val` - HubSpot access token to set.

`HasHubSpotRefreshToken()`

Checks if the company has a HubSpot refresh token.

Returns: `true` if a HubSpot refresh token is set, otherwise `false`.

`HubSpotRefreshToken()`

Gets the HubSpot refresh token for the company.

Returns: HubSpot refresh token.

`SetHubSpotRefreshToken($val)`

Sets the HubSpot refresh token for the company.

Parameters:

- `$val` - HubSpot refresh token to set.

`HasHubSpotExpiresTime()`

Checks if the company has a HubSpot expires time.

Returns: `true` if a HubSpot expires time is set, otherwise `false`.

`HubSpotExpiresTime()`

Gets the HubSpot expires time for the company.

Returns: HubSpot expires time.

`SetHubSpotExpiresTime($val)`

Sets the HubSpot expires time for the company.

Parameters:

- `$val` - HubSpot expires time to set.

`HasHubSpotPortalID()`

Checks if the company has a HubSpot portal ID.

Returns: `true` if a HubSpot portal ID is set, otherwise `false`.

`HubSpotPortalID()`

Gets the HubSpot portal ID for the company.

Returns: HubSpot portal ID.

`SetHubSpotPortalID($val)`

Sets the HubSpot portal ID for the company.

Parameters:

- `$val` - HubSpot portal ID to set.

Sendgrid

`SetSendGridApiKey($val)`

Sets the SendGrid API key for the company.

Parameters:

- `$val` - SendGrid API key to set.

`HasSendGridApiKey()`

Checks if the company has a SendGrid API key.

Returns: `true` if a SendGrid API key is set, otherwise `false`.

`SendGridApiKey()`

Gets the SendGrid API key for the company.

Returns: SendGrid API key.

`HasSendGridCompanyEmailDomainID()`

Checks if the company has a SendGrid company email domain ID.

Parameters: None

Returns: Boolean

`SendGridCompanyEmailDomainID()`

Gets the SendGrid company email domain ID associated with the company.

Parameters: None

Returns: ID (company_sendgrid_email_domain_id)

`SetSendGridCompanyEmailDomainID($val)`

Sets or updates the SendGrid company email domain ID for the company.

Parameters:

- `$val` - ID, the new company_sendgrid_email_domain_id.

Returns: None

Mailjet

`HasMailjetApiSecret()`

Checks if the company has a Mailjet API secret.

Returns: `true` if a Mailjet API secret is set, otherwise `false`.

`MailjetApiSecret()`

Gets the Mailjet API secret for the company.

Returns: Mailjet API secret.

`SetMailjetApiSecret($val)`

Sets the Mailjet API secret for the company.

Parameters:

- `$val` - Mailjet API secret to set.

`isMailjetApiSetUp()`

Checks if the company has set up Mailjet API with both API key and secret.

Returns: `true` if Mailjet API is set up, otherwise `false`.

`HasMailjetApiKey()`

Checks if the company has a Mailjet API key.

Returns: `true` if a Mailjet API key is set, otherwise `false`.

`MailjetApiKey()`

Gets the Mailjet API key for the company.

Returns: Mailjet API key.

`isValidMailjetSender()`

Checks if the company is a valid Mailjet sender.

Returns: `true` if a valid Mailjet sender.

`SetValidMailjetSender()`

Sets the company as a valid Mailjet sender.

`SetMailJetEmail($val)`

Sets the MailJet email for the company.

Parameters:

- `$val` - The value representing the MailJet email.

`HasMailJetEmail()`

Checks if the company has a MailJet email set.

Returns: Boolean value indicating whether the MailJet email is set.

`MailJetEmail()`

Gets the MailJet email for the company.

Returns: The MailJet email.

`SetMailjetApiKey($val)`

Sets the Mailjet API key for the company.

Parameters:

- `$val` - The value representing the Mailjet API key.

Nylas

`SetNylasClientID($val)`

Updates the Nylas Client ID for the current company.

Parameters:

- `$val` - Nylas Client ID value.

`HasNylasClientID()`

Checks if the company has a Nylas Client ID.

Returns: Boolean

`NylasClientID()`

Returns the Nylas Client ID of the company.

Returns: String

`HasNylasAPI()`

Checks if the company has both Nylas Client ID and Nylas Client Secret.

Returns: Boolean

`SetNylasClientSecret($val)`

Updates the Nylas Client Secret for the current company.

Parameters:

- `$val` - Nylas Client Secret value.

`HasNylasClientSecret()`

Checks if the company has a Nylas Client Secret.

Returns: Boolean

`NylasClientSecret()`

Returns the Nylas Client Secret of the company.

Returns: String

Twilio

`HasTwilioAccountSid()`

Checks if the company has a Twilio Account SID.

Returns: Boolean

`TwilioAccountSid()`

Returns the Twilio Account SID of the company.

Returns: String

`SetTwilioAccountSid($val)`

Updates the Twilio Account SID for the current company.

Parameters:

- `$val` - Twilio Account SID value.

`HasTwilioAuthToken()`

Checks if the company has a Twilio Auth Token.

Returns: Boolean

`TwilioAuthToken()`

Returns the Twilio Auth Token of the company.

Returns: String

`SetTwilioAuthToken($val)`

Updates the Twilio Auth Token for the current company.

Parameters:

- `$val` - Twilio Auth Token value.

`HasTwilioApiSid()`

Checks if the company has a Twilio API SID.

Returns: Boolean

`TwilioApiSid()`

Returns the Twilio API SID of the company.

Returns: String

`SetTwilioApiSid($val)`

Updates the Twilio API SID for the current company.

Parameters:

- `$val` - Twilio API SID value.

`TwilioTwiMLAppSID()`

Returns the Twilio TwiML App SID of the company.

Returns: String

`HasTwilioTwiMLAppSID()`

Checks if the company has a Twilio TwiML App SID.

Returns: Boolean

`SetTwilioTwiMLAppSID($val)`

Updates the Twilio TwiML App SID for the current company.

Parameters:

- `$val` - Twilio TwiML App SID value.

`HasTwilioApiSecret()`

Checks if the company has a Twilio API Secret.

Returns: Boolean

`TwilioApiSecret()`

Returns the Twilio API Secret of the company.

Returns: String

`SetTwilioApiSecret($val)`

Updates the Twilio API Secret for the current company.

Parameters:

- `$val` - Twilio API Secret value.

`TwilioAvailable()`

Checks if Twilio integration is available for the company.

Returns: Boolean

`CompanyVideoCallAvailable()`

Checks if video call functionality with Twilio is available for the company.

Returns: Boolean

`VideoCallAvailable()`

Checks if general video call functionality is available for the company.

Returns: Boolean

`SetNotificationsToCellphone($cellphone)`

Updates the Twilio phone number for notifications for the current company.

Parameters:

- `$cellphone` - Twilio phone number value.

`SMSNotification($text)`

Sends an SMS notification to the company's Twilio phone number if notifications are enabled.

Parameters:

- `$text` - Text of the SMS notification.

`HasCellphone()`

Checks if the company has a Twilio phone number.

Returns: Boolean

`Cellphone()`

Returns the Twilio phone number of the company.

Returns: String

`SetCellphone($cellphone)`

Updates the Twilio phone number for the current company.

Parameters:

- `$cellphone` - Twilio phone number value.

`PhoneForConversation()`

Returns the Twilio phone number for conversation purposes.

Returns: String

`SendNotificationsToCellphone()`

Returns whether notifications are sent to the company's Twilio phone number.

Returns: Boolean

`HasSendNotificationsToCellphone()`

Checks if notifications are sent to the company's Twilio phone number.

Returns: Boolean

`CheckTwilioPhone($twiliophone, $id_company=0)`

Checks if a Twilio phone number exists for a company.

Parameters:

- `$twiliophone` - Twilio phone number.
- `$id_company` - ID of the company (optional).

Returns: `true` if the Twilio phone number exists.

Pricing Plan

`HasPricingPlanID()`

Checks if the company has a pricing plan ID.

Returns: Boolean

`SetPricingPlan($val)`

Updates the pricing plan ID for the current company.

Parameters:

- `$val` - Integer value.

Invite Message

`InitialWWWInviteMessageID()`

Returns the initial message ID for the company's website invite.

Returns: Integer

`HasInitialWWWInviteMessageID()`

Checks if there is an initial message ID for the website invite.

Returns: Boolean

`SetInitialWWWInviteMessageID($id)`

Updates the initial message ID for the website invite.

Parameters:

- `$id` - Integer value.

Logo Image

`SetSystemLogoImage($filename, $erase_source_after_copying=true)`

Sets the system logo image for the company.

Parameters:

- `$filename` - Path to the image file.
- `$erase_source_after_copying` - Flag to erase the source after copying (default is `true`).

Returns: `true` if successful.

`UnsetSystemLogoImage()`

Unsets the system logo image for the company.

Returns: `true` if successful.

`SystemLogoImagePath()`

Returns the path to the system logo image.

`SystemLogoImageURL()`

Returns the URL of the system logo image.

`HasSystemLogoImageURL()`

Checks if the company has a system logo image.

Returns: `true` if the system logo image exists.

Customer Fields

`isCustomerFieldEnabled($field)`

Checks if a specific customer field is enabled.

Parameters:

- `$field` - Customer field.

Returns: `true` if the customer field is enabled.

`EnableCustomerField($field)`

Enables a specific customer field.

Parameters:

- `$field` - Customer field to enable.

`DisableCustomerField($field)`

Disables a specific customer field.

Parameters:

- `$field` - Customer field to disable.

`HasCustomerFormTemplate()`

Checks if the company has a customer form template.

Returns: `true` if a customer form template exists.

`CustomerFormTemplate()`

Returns the ID of the customer form template.

`SetCustomerFormTemplate($val)`

Sets the ID of the customer form template.

Parameters:

- `$val` - ID of the customer form template.

`isExpired()`

Checks if the company is expired.

Returns: `false`.

`MobileAppAuthorizationCode()`

Returns the mobile app authorization code.

`HasEmailFrom()`

Checks if the company has an email address.

Returns: `true` if an email address exists.

`EmailFrom()`

Returns the email address of the company.

`SetEmailFrom($val)`

Sets the email address for the company.

Parameters:

- `$val` - Email address.

`SetCompanyAppointmentReminder($val)`

Sets the company appointment reminder value.

Parameters:

- `$val` - The value to set for the company appointment reminder.

Returns: None

`HasCompanyAppointmentReminder()`

Checks if a company appointment reminder is set.

Parameters: None

Returns: `bool` - Indicates whether the company appointment reminder is set.

`CompanyAppointmentReminder()`

Gets the value of the company appointment reminder.

Parameters: None

Returns: The company appointment reminder value.

`InitialComplianceTagFirstName()`

Checks if the initial compliance message tag is set to 'first_name'.

Parameters: None

Returns: `bool` - Indicates whether the initial compliance tag is set to 'first_name'.

`InitialComplianceTagFirstLastName()`

Checks if the initial compliance message tag is set to 'first_and_last_name'.

Parameters: None

Returns: `bool` - Indicates whether the initial compliance tag is set to 'first_and_last_name'.

`InitialComplianceTagFirstAnonym()`

Checks if the initial compliance message tag is set to 'anonymous'.

Parameters: None

Returns: `bool` - Indicates whether the initial compliance tag is set to 'anonymous'.

`HasInitialComplianceMessageName()`

Checks if an initial compliance message name is set.

Parameters: None

Returns: `bool` - Indicates whether an initial compliance message name is set.

`InitialComplianceMessageName()`

Gets the value of the initial compliance message name.

Parameters: None

Returns: The initial compliance message name.

`SetInitialComplianceMessageName($val)`

Sets the initial compliance message name.

Parameters:

- `$val` - The value to set for the initial compliance message name.

Returns: None

`HasInitialComplianceMessageText()`

Checks if an initial compliance message text is set.

Parameters: None

Returns: `bool` - Indicates whether an initial compliance message text is set.

`InitialComplianceMessageText()`

Gets the value of the initial compliance message text.

Parameters: None

Returns: The initial compliance message text.

`SetInitialComplianceMessageText($val)`

Sets the initial compliance message text.

Parameters:

- `$val` - The value to set for the initial compliance message text.

Returns: None

`InitialAssetID()`

Gets the initial asset ID as an integer.

Parameters: None

Returns: `int` - The initial asset ID.

`HasInitialAssetID()`

Checks if an initial asset ID is set.

Parameters: None

Returns: `bool` - Indicates whether an initial asset ID is set.

`SetInitialAssetID($id)`

Sets the initial asset ID.

Parameters:

- `$id` - The value to set for the initial asset ID.

Returns: None

`BusinessWizardAvailable()`

Checks if the business wizard is available.

Parameters: None

Returns: `bool` - Indicates whether the business wizard is available.

`EnableBusinessWizard()`

Enables the business wizard.

Parameters: None

Returns: None

`DisableBusinessWizard()`

Disables the business wizard.

Parameters: None

Returns: None

`VcardName()`

Gets the value of the vCard name.

Parameters: None

Returns: The vCard name.

`HasVcardName()`

Checks if a vCard name is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard name is set.

`SetVcardName($link)`

Sets the vCard name.

Parameters:

- `$link` - The value to set for the vCard name.

Returns: None

`VcardPhone()`

Gets the value of the vCard phone.

Parameters: None

Returns: The vCard phone.

`HasVcardPhone()`

Checks if a vCard phone is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard phone is set.

`SetVcardPhone($val)`

Sets the vCard phone.

Parameters:

- `$val` - The value to set for the vCard phone.

Returns: None

`VcardEmail()`

Gets the value of the vCard email.

Parameters: None

Returns: The vCard email.

`HasVcardEmail()`

Checks if a vCard email is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard email is set.

`SetVcardEmail($val)`

Sets the vCard email.

Parameters:

- `$val` - The value to set for the vCard email.

Returns: None

`VcardWebsite()`

Gets the value of the vCard website.

Parameters: None

Returns: The vCard website.

`HasVcardWebsite()`

Checks if a vCard website is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard website is set.

`SetVcardWebsite($link)`

Sets the vCard website.

Parameters:

- `$link` - The value to set for the vCard website.

Returns: None

`VcardAddress()`

Gets the value of the vCard address.

Parameters: None

Returns: The vCard address.

`HasVcardAddress()`

Checks if a vCard address is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard address is set.

`SetVcardAddress($val)`

Sets the vCard address.

Parameters:

- `$val` - The value to set for the vCard address.

Returns: None

`VcardAddress2()`

Gets the value of the second vCard address line.

Parameters: None

Returns: The second vCard address line.

`HasVcardAddress2()`

Checks if a second vCard address line is set.

Parameters: None

Returns: `bool` - Indicates whether a second vCard address line is set.

`SetVcardAddress2($val)`

Sets the second vCard address line.

Parameters:

- `$val` - The value to set for the second vCard address line.

Returns: None

`VcardCity()`

Gets the value of the vCard city.

Parameters: None

Returns: The vCard city.

`HasVcardCity()`

Checks if a vCard city is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard city is set.

`SetVcardCity($val)`

Sets the vCard city.

Parameters:

- `$val` - The value to set for the vCard city.

Returns: None

`VcardZip()`

Gets the value of the vCard ZIP code.

Parameters: None

Returns: The vCard ZIP code.

`HasVcardZip()`

Checks if a vCard ZIP code is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard ZIP code is set.

`SetVcardZip($val)`

Sets the vCard ZIP code.

Parameters:

- `$val` - The value to set for the vCard ZIP code.

Returns: None

`VcardState()`

Gets the value of the vCard state.

Parameters: None

Returns: The v

Card state.

`HasVcardState()`

Checks if a vCard state is set.

Parameters: None

Returns: `bool` - Indicates whether a vCard state is set.

`SetVcardState($val)`

Sets the vCard state.

Parameters:

- `$val` - The value to set for the vCard state.

Returns: None

`FacebookURL()`

Gets the value of the Facebook URL.

Parameters: None

Returns: The Facebook URL.

`HasFacebookURL()`

Checks if a Facebook URL is set.

Parameters: None

Returns: `bool` - Indicates whether a Facebook URL is set.

`SetFacebookURL($link)`

Sets the Facebook URL.

Parameters:

- `$link` - The value to set for the Facebook URL.

Returns: None

`TwitterURL()`

Gets the value of the Twitter URL.

Parameters: None

Returns: The Twitter URL.

`HasTwitterURL()`

Checks if a Twitter URL is set.

Parameters: None

Returns: `bool` - Indicates whether a Twitter URL is set.

`SetTwitterURL($link)`

Sets the Twitter URL.

Parameters:

- `$link` - The value to set for the Twitter URL.

Returns: None

`LinkedinURL()`

Gets the value of the LinkedIn URL.

Parameters: None

Returns: The LinkedIn URL.

`HasLinkedinURL()`

Checks if a LinkedIn URL is set.

Parameters: None

Returns: `bool` - Indicates whether a LinkedIn URL is set.

`SetLinkedinURL($link)`

Sets the LinkedIn URL.

Parameters:

- `$link` - The value to set for the LinkedIn URL.

Returns: None

`YoutubeURL()`

Gets the value of the YouTube URL.

Parameters: None

Returns: The YouTube URL.

`HasYoutubeURL()`

Checks if a YouTube URL is set.

Parameters: None

Returns: `bool` - Indicates whether a YouTube URL is set.

`SetYoutubeURL($link)`

Sets the YouTube URL.

Parameters:

- `$link` - The value to set for the YouTube URL.

Returns: None

`InstagramURL()`

Gets the value of the Instagram URL.

Parameters: None

Returns: The Instagram URL.

`HasInstagramURL()`

Checks if an Instagram URL is set.

Parameters: None

Returns: `bool` - Indicates whether an Instagram URL is set.

`SetInstagramURL($link)`

Sets the Instagram URL.

Parameters:

- `$link` - The value to set for the Instagram URL.

Returns: None

`NewMessageFromCustomerNotification($customer, $message)`

Sends a notification to the employee about a new message from a customer.

Parameters:

- `$customer` - The customer object.
- `$message` - The message content.

Returns: None

Metadata

`LoadMetaData()`

Loads metadata for the company object.

Parameters: None

Returns: None

`GetMetaData(key)`

Retrieves the value of a specific metadata key for the company.

Parameters:

- `key` - Key to retrieve metadata for.

Returns: Metadata value for the specified key.

`SetMetaData(key, value, use_empty_as_null)`

Sets or updates the value of a metadata key for the company.

Parameters:

- `key` - Key to set or update.
- `value` - Value to set. Use NULL to delete metadata.
- `use_empty_as_null` - Use NULL when the value is empty.

Returns: None

`Create()`

Creates a new company object.

Parameters: None

Returns: New company object.

`isCustomHasThankYouPage()`

Checks if the company has a custom thank you page mode.

Parameters: None

Returns: Boolean indicating whether the company has a custom thank you page mode.

`HasThankYouTitle()`

Checks if the company has a thank you page title.

Parameters: None

Returns: Boolean indicating whether the company has a thank you page title.

`HasThankYouText()`

Checks if the company has a thank you page text.

Parameters: None

Returns: Boolean indicating whether the company has a thank you page text.

`ThankYouTitle()`

Retrieves the thank you page title for the company.

Parameters: None

Returns: Thank you page title.

`ThankYouText()`

Retrieves the thank you page text for the company.

Parameters: None

Returns: Thank you page text.

`GetThankYouPageMode()`

Retrieves the thank you page mode for the company.

Parameters: None

Returns: Thank you page mode.

`GetThankYouPageTitle()`

Retrieves the thank you page title for the company.

Parameters: None

Returns: Thank you page title.

`GetThankYouPageText()`

Retrieves the thank you page text for the company.

Parameters: None

Returns: Thank you page text.

`SetThankYouPageMode(type)`

Sets the thank you page mode for the company.

Parameters:

- `type` - Type of thank you page.

Returns: None

`SetThankYouPageTitle(title, val)`

Sets the thank you page title for the company.

Parameters:

- `title` - Title for the thank you page.
- `val` - Boolean value.

Returns: None

`SetThankYouPageText(text, val)`

Sets the thank you page text for the company.

Parameters:

- `text` - Text for the thank you page.
- `val` - Boolean value.

Returns: None

`HasLeadSignupLogoImageURL()`

Checks if the company has a lead signup logo image URL.

Parameters: None

Returns: Boolean indicating whether the company has a lead signup logo image URL.

`LeadSignupLogoImageURL()`

Retrieves the lead signup logo image URL for the company.

Parameters: None

Returns: Lead signup logo image URL.

`HasLogo()`

Checks if the company has a logo.

Parameters: None

Returns: Boolean indicating whether the company has a logo.

`ShowCustomLogo(): bool`

Returns a boolean indicating whether to show the custom logo.

Parameters: None

Returns: Boolean value.

`Logo()`

Retrieves the logo URL for the company.

Parameters: None

Returns: Logo URL.

`SetCustomLogoImage(filename, erase_source_after_copying)`

Sets the custom logo image for the company.

Parameters:

- `filename` - Path to the image file.

- `erase_source_after_copying` - Boolean indicating whether to erase the source after copying.

Returns: Boolean indicating success or failure.

`isEnabledReferralSignupField(field)`

Checks if the specified referral signup field is enabled for the company.

Parameters:

- `field` - Referral signup field.

Returns: Boolean indicating whether the field is enabled.

`EnableLeadSignupField(field)`

Enables the specified lead signup field for the company.

Parameters:

- `field` - Lead signup field to enable.

Returns: None

`DisableLeadSignupField(field)`

Disables the specified lead signup field for the company.

Parameters:

- `field` - Lead signup field to disable.

Returns: None

`SetLeadSignupThankyouPageImage(filename, erase_source_after_copying)`

Sets the lead signup thank you page image for the company.

Parameters:

- `filename` - Path to the image file.
- `erase_source_after_copying` - Boolean indicating whether to erase the source after copying.

Returns: Boolean indicating success or failure.

`LeadSignupThankyouPageImageURL()`

Retrieves the lead signup thank you page image URL for the company.

Parameters: None

Returns: Lead signup thank you page image URL.

`HasLeadSignupThankyouPageImageURL()`

Checks if the company has a lead signup thank you page image URL.

Parameters: None

Returns: Boolean indicating whether the company has a lead signup thank you page image URL.

`StripeSecretKey()`

Retrieves the Stripe secret key for the company.

Parameters: None

Returns: Stripe secret key.

`HasStripePublicKey()`

Checks if the company has a Stripe public key.

Parameters: None

Returns: Boolean indicating whether the company has a Stripe public key.

`HasStripeWebhooksEndpointSecret()`

Checks if the company has a Stripe webhooks endpoint secret.

Parameters: None

Returns: Boolean indicating whether the company has a Stripe webhooks endpoint secret.

`HasStripeSettings()`

Checks if the company has all required Stripe settings.

Parameters: None

Returns: Boolean indicating whether the company has all required Stripe settings.

`HasStripeSecretKey()`

Checks if the company has a Stripe secret key.

Parameters: None

Returns: Boolean indicating whether the company has a Stripe secret key.

`StripePublicKey()`

Retrieves the Stripe public key for the company.

Parameters: None

Returns: Stripe public key.

`StripeWebhooksEndpointSecret()`

Retrieves the Stripe webhooks endpoint secret for the company.

Parameters: None

Returns: Stripe webhooks endpoint secret.

`CompanyPipeline()`

Retrieves the company pipeline for the company.

Parameters: None

Returns: Company pipeline.

`isOrganizationBasedPipeline()`

Checks if the company's company pipeline is organization-based.

Parameters: None

Returns: Boolean indicating whether the company pipeline is organization-based.

`isCustomerBasedPipeline()`

Checks if the company's company pipeline is customer-based.

Parameters: None

Returns: Boolean indicating whether the company pipeline is customer-based.

`CurrentPipeline()`

Retrieves the current pipeline for the company.

Parameters: None

Returns: Current pipeline.

`SetCompanyPipeline(val)`

Sets the company pipeline for the company.

Parameters:

- `val` - Value to set.

Returns: None

Stripe

`SetStripeSecretKey(val)`

Sets the Stripe secret key for the company.

Parameters:

- `val` - Value to set.

Returns: None

`SetStripePublicKey(val)`

Sets the Stripe public key for the company.

Parameters

:

- `val` - Value to set.

Returns: None

`SetStripeWebhooksEndpointSecret(val)`

Sets the Stripe webhooks endpoint secret for the company.

Parameters:

- `val` - Value to set.

Returns: None

`isStripeTestMode()`

Checks if the company is in Stripe test mode.

Parameters: None

Returns: Boolean indicating whether the company is in Stripe test mode.

`isEnabledSaveToPurchasesTestMode()`

Checks if saving to purchases is enabled in test mode for the company.

Parameters: None

Returns: Boolean indicating whether saving to purchases is enabled in test mode.

`EnableSaveToPurchasesTestMode()`

Enables saving to purchases in test mode for the company.

Parameters: None

Returns: None

`DisableSaveToPurchasesTestMode()`

Disables saving to purchases in test mode for the company.

Parameters: None

Returns: None

`SetContentLabel($label)`

Sets the content label for the company.

Parameters:

- `$label` - Content label to be set.

Returns: None

`ContentLabel()`

Returns the content label associated with the company.

Parameters: None

Returns: Content label value.

`SetUserID($val)`

Sets the user ID for the company.

Parameters:

- `$val` - User ID to be set.

Returns: None

`UserID()`

Returns the user ID associated with the company.

Parameters: None

Returns: User ID value.

`HasUserID()`

Checks if the company has a user ID.

Parameters: None

Returns: Boolean indicating whether the company has a user ID.

`isPaymentTypePostMoney()`

Checks if the payment type for posts is set to 'money'.

Parameters: None

Returns: Boolean indicating the payment type for posts.

`isPaymentTypePostPoints()`

Checks if the payment type for posts is set to 'points'.

Parameters: None

Returns: Boolean indicating the payment type for posts.

`SetPaymentTypePost($tag)`

Sets the payment type for posts.

Parameters:

- `$tag` - Payment type for posts ('money' or 'points').

Returns: None

`SetPaymentTypeLeads($tag)`

Sets the payment type for leads.

Parameters:

- `$tag` - Payment type for leads ('money' or 'points').

Returns: None

`isPaymentTypeLeadsMoney()`

Checks if the payment type for leads is set to 'money'.

Parameters: None

Returns: Boolean indicating the payment type for leads.

`isPaymentTypeLeadsPoints()`

Checks if the payment type for leads is set to 'points'.

Parameters: None

Returns: Boolean indicating the payment type for leads.

`LeadToCustomerPrice()`

Returns the lead to customer price.

Parameters: None

Returns: Lead to customer price value.

`SetLeadToCustomerPrice($price)`

Sets the lead to customer price.

Parameters:

- `$price` - Lead to customer price to be set.

Returns: None

`isLeadsSectionInMemberAreaVisible()`

Checks if the leads section is visible in the member area.

Parameters: None

Returns: Boolean indicating the visibility of the leads section.

`SetLeadsSectionInMemberAreaVisibility($visible)`

Sets the visibility of the leads section in the member area.

Parameters:

- `$visible` - Boolean indicating whether the leads section should be visible.

Returns: None

`CloneSystemTemplates()`

Clones system templates associated with the company.

Parameters: None

Returns: None

`SendEmailFromCompany($subject, $message, $email, $company = '')`

Sends an email from the company to the specified email address.

Parameters:

- `$subject` - Subject of the email.
- `$message` - Content of the email.
- `$email` - Email address to send the email to.
- `$company` - (Optional) Company object (default is current company).

Returns: Boolean indicating the success of email sending.

`SendNotificationSMS($text)`

Sends a notification SMS to the company.

Parameters:

- `$text` - Text content of the SMS.

Returns: None

`checkCompanyWithEmail($email)`

Static method to check a company with the specified email.

Parameters:

- `$email` - Email address to check.

Returns: An instance of the company if found, otherwise false.

Labels

`SetLabelForStage($id, $val)`

Sets the label for a specific stage.

Parameters:

- `$id` - ID of the stage.
- `$val` - New label for the stage.

Returns: None

`SetLabelForStageTagByStatus($status_tag, $stage_tag, $val)`

Sets the label for a stage based on status and tag.

Parameters:

- `$status_tag` - Status tag (e.g., 'customers', 'opportunities', 'leads').
- `$stage_tag` - Stage tag.
- `$val` - New label for the stage.

Returns: None

`GetLabelForStage($id)`

Gets the label for a specific stage.

Parameters:

- `$id` - ID of the stage.

Returns: Label for the specified stage.

`LabelForCustomersStatus()`

Gets the label for customers' status.

Returns: Label for customers' status.

`LabelForOpportunitiesStatus()`

Gets the label for opportunities' status.

Returns: Label for opportunities' status.

`LabelForLeadsStatus()`

Gets the label for leads' status.

Returns: Label for leads' status.

`LabelForReceivedStage()`

Gets the label for the received stage.

Returns: Label for the received stage.

`LabelForContactStage()`

Gets the label for the contact stage.

Returns: Label for the contact stage.

`LabelForAppointmentStage()`

Gets the label for the appointment stage.

Returns: Label for the appointment stage.

`LabelForSoldStage()`

Gets the label for the sold stage.

Returns: Label for the sold stage.

`LabelForLostStage()`

Gets the label for the lost stage.

Returns: Label for the lost stage.

`LabelForRefusedStage()`

Gets the label for the refused stage.

Returns: Label for the refused stage.

`LabelForReceivedOpportunityStage()`

Gets the label for the received opportunity stage.

Returns: Label for the received opportunity stage.

`LabelForContactOpportunityStage()`

Gets the label for the contact opportunity stage.

Returns: Label for the contact opportunity stage.

`LabelForAppointmentOpportunityStage()`

Gets the label for the appointment opportunity stage.

Returns: Label for the appointment opportunity stage.

`LabelForSoldOpportunityStage()`

Gets the label for the sold opportunity stage.

Returns: Label for the sold opportunity stage.

`LabelForLostOpportunityStage()`

Gets the label for the lost opportunity stage.

Returns: Label for the lost opportunity stage.

`LabelForRefusedOpportunityStage()`

Gets the label for the refused opportunity stage.

Returns: Label for the refused opportunity stage.

`LabelForReceivedLeadStage()`

Gets the label for the received lead stage.

Returns: Label for the received lead stage.

`LabelForContactLeadStage()`

Gets the label for the contact lead stage.

Returns: Label for the contact lead stage.

`LabelForAppointmentLeadStage()`

Gets the label for the appointment lead stage.

Returns: Label for the appointment lead stage.

`LabelForSoldLeadStage()`

Gets the label for the sold lead stage.

Returns: Label for the sold lead stage.

`LabelForLostLeadStage()`

Gets the label for the lost lead stage.

Returns: Label for the lost lead stage.

`LabelForRefusedLeadStage()`

Gets the label for the refused lead stage.

Returns: Label for the refused lead stage.

`LabelForSupportActive()`

Returns the label for the active support status.

`LabelForSupportClosed()`

Returns the label for the closed support status.

`SetLabelForSupportActive($label)`

Sets the label for the active support status.

Parameters:

- `$label` - Label for the active support status.

`SetLabelForSupportClose($label)`

Sets the label for the closed support status.

Parameters:

- `$label` - Label for the closed support status.

`SetLabelForCustomersStatus($label)`

Sets the label for the customers' status.

Parameters:

- `$label` - Label for the customers' status.

`SetLabelForStatus($status, $label)`

Sets the label for a specific status.

Parameters:

- `$status` - Status (e.g., "opportunities").
- `$label` - Label for the status.

`SetLabelForOpportunitiesStatus($label)`

Sets the label for the opportunities' status.

Parameters:

- `$label` - Label for the opportunities' status.

`SetLabelForLeadsStatus($label)`

Sets the label for the leads' status.

Parameters:

- `$label` - Label for the leads' status.

`LabelForCustomer()`

Returns the label for a single customer.

`LabelForCustomers()`

Returns the label for multiple customers.

`SetLabelForCustomer($label)`

Sets the label for a single customer.

Parameters:

- `$label` - Label for a single customer.

`SetLabelForCustomers($label)`

Sets the label for multiple customers.

Parameters:

- `$label` - Label for multiple customers.

`LabelForReferral()`

Returns the label for a referral.

`SetLabelForReferral($label)`

Sets the label for a referral.

Parameters:

- `$label` - Label for a referral.

`LabelForReferrals()`

Returns the label for multiple referrals.

`SetLabelForReferrals($label)`

Sets the label for multiple referrals.

Parameters:

- `$label` - Label for multiple referrals.

`LabelForLead()`

Returns the label for a lead.

`SetLabelForLead($label)`

Sets the label for a lead.

Parameters:

- `$label` - Label for a lead.

`LabelForLeads()`

Returns the label for multiple leads.

`SetLabelForLeads($label)`

Sets the label for multiple leads.

Parameters:

- `$label` - Label for multiple leads.

`GetStagesTitlesForStatus(string $status_tag): array`

Gets an array of stage titles for a given status tag.

Parameters:

- `$status_tag` - Status tag.

Returns: Array of stage titles.

`PipelineTypeValuesArray(): array`

Gets an array of pipeline type values.

Returns: Array of pipeline type values.

`PipelineTypeTitlesArray(): array`

Gets an array of pipeline type titles.

Returns: Array of pipeline type titles.

`PipelineTitleForStatus(string $status): string`

Gets the pipeline title for a given status.

Parameters:

- `$status` - Status tag.

Returns: Pipeline title.

`InitPipelineTagForTitle(string $status): string`

Initializes the pipeline tag based on the pipeline title.

Parameters:

- `$status` - Pipeline title.

Returns: Pipeline tag.

Pipelines

`InitStageTagForTitle(string $stage): array`

Initializes the stage tag and status based on the stage title.

Parameters:

- `$stage` - Stage title.

Returns: Array with 'stage' and 'status' keys.

`PipelineTitleForStatusStage(string $status_tag, string $stage_tag): string`

Gets the pipeline title for a given status and stage tag.

Parameters:

- `$status_tag` - Status tag.
- `$stage_tag` - Stage tag.

Returns: Pipeline title.

Domains

`SetEmailDomain($val)`

Updates the email domain value.

Parameters:

- `$val` - The new email domain value.

`HasEmailDomain()`

Checks if the company has an email domain.

Returns:

- `boolean` - `true` if the company has an email domain, `false` otherwise.

`EmailDomain()`

Returns the email domain of the company.

Returns:

- `string` - The email domain.

`SetCompanyEmailDnsID($val)`

Sets the DNS ID for the company email.

Parameters:

- `$val` - The new DNS ID.

`HasCompanyEmailDnsID()`

Checks if the company has a company email DNS ID.

Returns:

- `boolean` - `true` if the company has a company email DNS ID, `false` otherwise.

`CompanyEmailDnsID()`

Returns the company email DNS ID.

Returns:

- `mixed` - The company email DNS ID.

`SetCompanyEmailDomainID($val)`

Sets the domain ID for the company email.

Parameters:

- `$val` - The new domain ID.

`HasCompanyEmailDomainID()`

Checks if the company has a company email domain ID.

Returns:

- `boolean` - `true` if the company has a company email domain ID, `false` otherwise.

`CompanyEmailDomainID()`

Returns the company email domain ID.

Returns:

- `mixed` - The company email domain ID.

`SetCompanyEmailDomain($val)`

Sets the domain for the company email.

Parameters:

- `$val` - The new domain.

`HasCompanyEmailDomain()`

Checks if the company has a company email domain.

Returns:

- `boolean` - `true` if the company has a company email domain, `false` otherwise.

`CompanyEmailDomain()`

Returns the company email domain.

Returns:

- `mixed` - The company email domain.

`SetCompanyEmailOwnerShipToken($val)`

Sets the ownership token for the company email.

Parameters:

- `$val` - The new ownership token.

`HasCompanyEmailOwnerShipToken()`

Checks if the company has a company email ownership token.

Returns:

- `boolean` - `true` if the company has a company email ownership token, `false` otherwise.

`CompanyEmailOwnerShipToken()`

Returns the company email ownership token.

Returns:

- `mixed` - The company email ownership token.

`SetCompanyEmailOwnerShipTokenRecordName($val)`

Sets the record name for the ownership token of the company email.

Parameters:

- `$val` - The new record name for the ownership token.

`HasCompanyEmailOwnerShipTokenRecordName()`

Checks if the company has a record name for the ownership token of the company email.

Returns:

- `boolean` - `true` if the company has a record name for the ownership token, `false` otherwise.

`CompanyEmailOwnerShipTokenRecordName()`

Returns the record name for the ownership token of the company email.

Returns:

- `mixed` - The record name for the ownership token.

`SetCompanyEmailSPFRecordValue($val)`

Sets the SPF record value for the company email.

Parameters:

- `$val` - The new SPF record value.

`HasCompanyEmailSPFRecordValue()`

Checks if the company has an SPF record value for the company email.

Returns:

- `boolean` - `true` if the company has an SPF record value, `false` otherwise.

`CompanyEmailSPFRecordValue()`

Returns the SPF record value for the company email.

Returns:

- `mixed` - The SPF record value.

`SetCompanyEmailDKIMRecordName($val)`

Sets the record name for the DKIM of the company email.

Parameters:

- `$val` - The new record name for the DKIM.

`HasCompanyEmailDKIMRecordName()`

Checks if the company has a record name for the DKIM of the company email.

Returns:

- `boolean` - `true` if the company has a record name for the DKIM, `false` otherwise.

`CompanyEmailDKIMRecordName()`

Returns the record name for the DKIM of the company email.

Returns:

- `mixed` - The record name for the DKIM.

`SetCompanyEmailDKIMRecordValue($val)`

Sets the record value for the DKIM of the company email.

Parameters:

- `$val` - The new record value for the DKIM.

`HasCompanyEmailDKIMRecordValue()`

Checks if the company has a record value for the DKIM of the company email.

Returns:

- `boolean` - `true` if the company has a record value for the DKIM, `false` otherwise.

`CompanyEmailDKIMRecordValue()`

Returns the record value for the DKIM of the company email.

Returns:

- `mixed` - The record value for the DKIM.

`SetCompanyEmailDKIMRecordStatus($val)`

Sets the status for the DKIM record of the company email.

Parameters:

- `$val` - The new status for the DKIM record.

`HasCompanyEmailDKIMRecordStatus()`

Checks if the company has a status for the DKIM record of the company email.

Returns:

- `boolean` - `true` if the company has a status for the DKIM record, `false` otherwise.

`CompanyEmailDKIMRecordStatus()`

Returns the status for the DKIM record of the company email.

Returns:

- `mixed` - The status for the DKIM record.

`SetCompanyEmailSPFRecordStatus($val)`

Sets the status for the SPF record of the company email.

Parameters:

- `$val` - The new status for the SPF record.

`HasCompanyEmailSPFRecordStatus()`

Checks if the company has a status for the SPF record of the company email.

Returns:

- `boolean` - `true` if the company has a status for the SPF record, `false` otherwise.

`CompanyEmailSPFRecordStatus()`

Returns the status for the SPF record of the company email.

Returns:

- `mixed` - The status for the SPF record.

Email

`SetCompanyEmailStatus($val)`

Sets the status for the company email.

Parameters:

- `$val` - The new status for the company email.

`HasCompanyEmailStatus()`

Checks if the company has a status for the company email.

Returns:

- `boolean` - `true` if

the company has a status for the company email, `false` otherwise.

`CompanyEmailStatus()`

Returns the status for the company email.

Returns:

- `mixed` - The status for the company email.

`isCompanyEmailActive()`

Checks if the company email is active.

Returns:

- `boolean` - `true` if the company email status is 'Active', `false` otherwise.

`SetCompanyEmailParseRouteID($val)`

Sets the parse route ID for the company email.

Parameters:

- `$val` - The new parse route ID for the company email.

`HasCompanyEmailParseRouteURL()`

Checks if the company has a parse route URL for the company email.

Returns:

- `boolean` - `true` if the company has a parse route URL, `false` otherwise.

`CompanyEmailParseRouteURL()`

Returns the parse route URL for the company email.

Returns:

- `mixed` - The parse route URL for the company email.

`HasCompanyEmailParseRouteID()`

Checks if the company has a parse route ID for the company email.

Returns:

- `boolean` - `true` if the company has a parse route ID, `false` otherwise.

`CompanyEmailParseRouteID()`

Returns the parse route ID for the company email.

Returns:

- `mixed` - The parse route ID for the company email.

`SetCompanyEmailParseRouteURL($val)`

Sets the parse route URL for the company email.

Support

`SetSupportEmailDomain($val)`

Sets the support email domain for the company.

Parameters:

- `$val` - The value representing the support email domain.

`HasSupportEmailDomain()`

Checks if the company has a support email domain set.

Returns: Boolean value indicating whether the support email domain is set.

`SupportEmailDomain()`

Gets the support email domain for the company.

Returns: The support email domain.

`HasSupportTicketPrefix()`

Checks if the company has a support ticket prefix set.

Returns: Boolean value indicating whether the support ticket prefix is set.

`SetSupportTicketPrefix($val)`

Sets the support ticket prefix for the company.

Parameters:

- `$val` - The value representing the support ticket prefix.

`SupportTicketPrefix()`

Gets the support ticket prefix for the company. If not set, returns the default value 'LM'.

Returns: The support ticket prefix.

`HasSupportTicketStartNumber()`

Checks if the company has a support ticket start number set.

Returns: Boolean value indicating whether the support ticket start number is set.

`SetSupportTicketStartNumber($val)`

Sets the support ticket start number for the company.

Parameters:

- `$val` - The value representing the support ticket start number.

`SupportTicketStartNumber()`

Gets the support ticket start number for the company. If not set, returns the default value '1'.

Returns: The support ticket start number.

`SetSupportEmailDomainID($val)`

Sets the ID of the support email domain for the company.

Parameters:

- `$val` - The value representing the support email domain ID.

`HasSupportEmailDomainID()`

Checks if the company has a support email domain ID set.

Returns: Boolean value indicating whether the support email domain ID is set.

`SupportEmailDomainID()`

Gets the ID of the support email domain for the company.

Returns: The support email domain ID.

`SetSupportEmailStatus($val)`

Sets the status of the support email for the company.

Parameters:

- `$val` - The value representing the support email status.

`HasSupportEmailStatus()`

Checks if the company has a support email status set.

Returns: Boolean value indicating whether the support email status is set.

`SupportEmailStatus()`

Gets the status of the support email for the company.

Returns: The support email status.

`isSupportEmailActive()`

Checks if the support email is active for the company.

Returns: Boolean value indicating whether the support email is active.

`isSupportEnabled()`

Checks if support is enabled for the company.

Returns: `true` if support is enabled, otherwise `false`.

`GetSupportStatusesTitles()`

Gets an array of support statuses titles.

Returns: Array of support statuses titles.

`GetSupportStatuseIDs()`

Gets an array of support status IDs.

Returns: Array of support status IDs.

`SetSupportStatus($bool_val)`

Sets the status of support for the company.

Parameters:

- `$bool_val` - Boolean value to set for support status.

`SetSupportEmailDnsID($val)`

Sets the DNS ID of the support email for the company.

Parameters:

- `$val` - The value representing the support email DNS ID.

`HasSupportEmailDnsID()`

Checks if the company has a support email DNS ID set.

Returns: Boolean value indicating whether the support email DNS ID is set.

`SupportEmailDnsID()`

Gets the DNS ID of the support email for the company.

Returns: The support email DNS ID.

`SetSupportEmailOwnerShipToken($val)`

Sets the ownership token of the support email for the company.

Parameters:

- `$val` - The value representing the support email ownership token.

`HasSupportEmailOwnerShipToken()`

Checks if the company has a support email ownership token set.

Returns: Boolean value indicating whether the support email ownership token is set.

`SupportEmailOwnerShipToken()`

Gets the ownership token of the support email for the company.

Returns: The support email ownership token.

`SetSupportEmailOwnerShipTokenRecordName($val)`

Sets the record name of the ownership token for the support email.

Parameters:

- `$val` - The value representing the record name of the ownership token.

`HasSupportEmailOwnerShipTokenRecordName()`

Checks if the company has a record name set for the support email ownership token.

Returns: Boolean value indicating whether the record name for the ownership token is set.

`SupportEmailOwnerShipTokenRecordName()`

Gets the record name of the ownership token for the support email.

Returns: The record name of the ownership token.

`SetSupportEmailParseRouteID($val)`

Sets the parse route ID for the support email.

Parameters:

- `$val` - The value representing the parse route ID for the support email.

`HasSupportEmailParseRouteID()`

Checks if the company has a parse route ID set for the support email.

Returns: Boolean value indicating whether the parse route ID for the support email is set.

`SupportEmailParseRouteID()`

Gets the parse route ID for the support email.

Returns: The parse route ID for the support email.

`SetSupportEmailParseRouteURL($val)`

Sets the parse route URL for the support email.

Parameters:

- `$val` - The value representing the parse route URL for the support email.

`HasSupportEmailParseRouteURL()`

Checks if the company has a parse route URL set for the support email.

Returns: Boolean value indicating whether the parse route URL for the support email is set.

`SupportEmailParseRouteURL()`

Gets the parse route URL for the support email.

Returns: The parse route URL for the support email.

`SetSupportEmail($val)`

Updates the support email for the company.

Parameters:

- `$val` - The value representing the updated support email.

`HasSupportEmail()`

Checks if the company has a support email set.

Returns: Boolean value indicating whether the support email is set.

`SupportEmail()`

Gets the support email for the company.

Returns: The support email.

`HasSupportDepartments()`

Checks if the company has support departments.

Parameters: None

Returns: Boolean indicating the presence of support departments.

`LabelForHighPriorityStatus()`

Gets the label for high priority status.

Returns: Label for high priority status.

`LabelForLowPriorityStatus()`

Gets the label for low priority status.

Returns: Label for low priority status.

`LabelForMediumPriorityStatus()`

Gets the label for medium priority status.

Returns: Label for medium priority status.

`GetSupportPriorityTitle(string $priority_value)`

Returns the label for a given support priority value.

Parameters:

- `priority_value` - The support priority value.

Returns:

- `string` - The label corresponding to the support priority.

`GetSupportPriorityTitlesArray(): array`

Returns an array of support priority labels for all available support priority values.

Returns:

- `array` - An array containing support priority labels.

`SetLabelForSupportPriority($id, $val)`

Sets the label for a specific support priority.

Parameters:

- `$id` - The support priority ID.
- `$val` - The new label for the support priority.

Knowledgebase

`SetKnowledgebaseDomain($val)`

Updates the knowledge base domain for the company.

Parameters:

- `$val` - The value representing the updated knowledge base domain.

`HasKnowledgebaseDomain()`

Checks if the company has a knowledge base domain set.

Returns: Boolean value indicating whether the knowledge base domain is set.

`KnowledgebaseDomain()`

Gets the knowledge base domain for the company.

Returns: The knowledge base domain.

`SetKnowledgebaseDomainStatus($val)`

Sets the status of the knowledge base domain for the company.

Parameters:

- `$val` - The value representing the knowledge base domain status.

`HasKnowledgebaseDomainStatus()`

Checks if the company has a knowledge base domain status set.

Returns: Boolean value indicating whether the knowledge base domain status is set.

`KnowledgebaseDomainStatus()`

Gets the status of the knowledge base domain for the company.

Returns: The knowledge base domain status.

Favicon

`HasCustomDomainFavicon()`

Checks whether the custom domain favicon exists for the company.

Parameters: None

Returns: Boolean

`CustomDomainFaviconURL()`

Generates the URL for the custom domain favicon.

Parameters: None

Returns: String (URL)

`RemoveCustomDomainFavicon()`

Removes the custom domain favicon file.

Parameters: None

Returns: None

`SetCustomDomainFavicon($filename, $erase_source_after_copying)`

Sets the custom domain favicon using the provided filename.

Parameters:

- `$filename` - String, the path to the favicon file.
- `$erase_source_after_copying` - Boolean, whether to erase the source file after copying.

Returns: Boolean (true if successful, false otherwise)

Google API

`HasGoogleApiKey()`

Checks if the company has a Google Places API key.

Parameters: None

Returns: Boolean

`GoogleApiKey()`

Gets the Google Places API key associated with the company.

Parameters: None

Returns: String (API key)

`SetGoogleApiKey($val)`

Sets or updates the Google Places API key for the company.

Parameters:

- `$val` - String, the new API key.

Returns: None

ProxyCurl

`HasProxyCurlApiKey()`

Checks if the company has a ProxyCurl API key.

Parameters: None

Returns: Boolean

`ProxyCurlApiKey()`

Gets the ProxyCurl API key associated with the company.

Parameters: None

Returns: String (API key)

BuiltWith API

`HasBuiltWithApiKey()`

Checks if the company has a BuiltWith API key.

Parameters: None

Returns: Boolean

`BuiltWithApiKey()`

Gets the BuiltWith API key associated with the company.

Parameters: None

Returns: String (API key)

Project Management

`isProjectManagementEnabled()`

Checks if project management is enabled for the company.

Returns: `true` if project management is enabled, otherwise `false`.

`SetProjectManagementStatus($bool_val)`

Sets the status of project management for the company.

Parameters:

- `$bool_val` - Boolean value to set for project management status.

Company List class

- Full class name: `\GS\Company\CompanyList`
- Legacy alias (shouldn't be used in a new code): ~~`\TCompanyList`~~

Information:

- Type: ORM Model List
- DB Table: `companies`
- List Items: \GS\Company\Company

`SetFilterNotExpired()`

Sets a filter to retrieve only records with expiration dates that are not expired.

Parameters: None

Returns: None

`SetFilterBusinessType($val)`

Sets a filter based on the business type ID to retrieve records with a specific business type.

Parameters:

- `$val` - Business type ID (integer)

Returns: None

Customer class

- Full class name: `\GS\Customer\Customer`
- Legacy alias (shouldn't be used in a new code): ~~`\TCustomer`~~

Information:

- Type: ORM Model
- DB Table: `customers`

`\GS\Customer\Customer` has the responsibility for manipulating a single customer record in a database. It contains additional methods for setting tags, sending messages, etc.

Table of Contents

Initializers

`initWithPhonesArray($phones, $id_company)`

Initializes a customer object based on an array of phone numbers.

Parameters:

- `$phones` - Array of phone numbers with associated country codes.
- `$id_company` - Company ID.

Returns:

- An instance of the `Customer` class if a customer is found, otherwise an empty string.

`getByEmail($email)`

Gets customer information based on the provided email.

Parameters:

- `$email` - Email address of the customer.

Returns:

- Customer information if the email is found, otherwise `false`.

`initWithSupportContact($supportcontact)`

Initializes a customer object based on a support contact.

Parameters:

- `$supportcontact` - An instance of the `TSupportContact` class.

Returns:

- An instance of the `Customer` class if a customer is found, otherwise `false`.

`initWithCustomerPhoneNotDeleted($phone, $id_company, $country)`

Initializes a customer object based on a phone number, company ID, and optional country code.

Parameters:

- `$phone` - Phone number.
- `$id_company` - Company ID.
- `$country` - Optional country code.

Returns:

- An instance of the `Customer` class.

`initWithCustomerEmailNotDeleted($email, $id_company)`

Initializes a customer object based on an email and company ID.

Parameters:

- `$email` - Email address.
- `$id_company` - Company ID.

Returns:

- An instance of the `Customer` class.

`initWithCustomerPhone($phone, $id_company)`

(Deprecated) Initializes a customer object based on a phone number and company ID.

Parameters:

- `$phone` - Phone number.
- `$id_company` - Company ID.

Returns:

- An instance of the `Customer` class.

`initWithCustomerEmail($email, $id_company)`

Initializes a customer object based on an email and company ID.

Parameters:

- `$email` - Email address.
- `$id_company` - Company ID.

Returns:

- An instance of the `Customer` class.

`withID($id)`

Creates an instance of the class with a specific ID.

Parameters:

- `$id` - Customer ID.

Returns:

- An instance of the `Customer` class.

`UsingCache($id)`

Creates an instance of the class using cache for a specific ID.

Parameters:

- `$id` - Customer ID.

Returns:

- An instance of the `Customer` class.

Create

`CreateByStaff($company, $first_name, $last_name, $cellphone, $email)`

Creates a new customer initiated by staff.

Parameters:

- `company` - Company object.
- `first_name` - First name of the customer.
- `last_name` - Last name of the customer.
- `cellphone` - Customer's cellphone number.
- `email` - Customer's email address.

Returns: Created customer object.

`CreateByImport($company, $first_name, $last_name, $cellphone, $email, $address, $city, $state, $zip, $country)`

Creates a new customer initiated by import. Static method.

Parameters:

- `company` - Company object.
- `first_name` - First name of the customer.
- `last_name` - Last name of the customer.
- `cellphone` - Customer's cellphone number.
- `email` - Customer's email address.
- `address` - Customer's address.
- `city` - Customer's city.
- `state` - Customer's state.
- `zip` - Customer's ZIP code.
- `country` - Customer's country.

Returns: Created customer object.

`WebCreate($company, $employee=null, $first_name='', $last_name='', $cellphone='', $email='', $address='', $city='', $state='', $zip='', $source='', $offer=NULL)`

Creates a customer using web-based information. Static method.

Parameters:

- `$company` - TCompany object associated with the customer.
- `$employee` - TEmployee object associated with the customer.
- `$first_name` - First name of the customer.
- `$last_name` - Last name of the customer.
- `$cellphone` - Cellphone number of the customer.
- `$email` - Email address of the customer.
- `$address` - Address of the customer.
- `$city` - City of the customer.
- `$state` - State of the customer.
- `$zip` - Zip code of the customer.
- `$source` - Source of the customer.
- `$offer` - Offer associated with the customer.

`CreateByInvitedCustomerWWW($company, $first_name, $last_name, $cellphone)`

Creates a customer based on information from an invited customer. Static method.

Parameters:

- `$company` - TCompany object associated with the customer.
- `$first_name` - First name of the customer.
- `$last_name` - Last name of the customer.
- `$cellphone` - Cellphone number of the customer.

Returns: TCustomer object.

`ONCreateEvent()`

Event triggered when a new customer is created.

Parameters: None

Returns: No return value.

`Create($company)`

Creates a customer. Static method.

Parameters:

- `$company` - TCompany object associated with the customer.

Emails Sending

`SendEmailFromCompany($subject, $message, $scheduletime, $writelog, $employee, $campaign, $campaign_item, $bulk, $id_campaign_schedule, $email_id, $cc, $bcc)`

Sends an email from the company to the customer.

Parameters:

- `$subject` - Email subject.
- `$message` - Email message.
- `$scheduletime` - Scheduled time for sending the email.
- `$writelog` - Boolean flag indicating whether to write a log.
- `$employee` - Employee ID or object.
- `$campaign` - Campaign ID.
- `$campaign_item` - Campaign item ID.
- `$bulk` - Boolean flag indicating if the email is part of a bulk operation.
- `$id_campaign_schedule` - Campaign schedule ID.
- `$email_id` - Email ID.
- `$cc` - CC recipients.
- `$bcc` - BCC recipients.

Returns:

- `true` if the email is sent successfully, otherwise `false`.

`SendEmailFromEmployee($subject, $message, $scheduletime, $writelog, $employee, $campaign, $campaign_item, $email_id, $cc, $bcc)`

Sends an email from an employee to the customer.

Parameters:

- `$subject` - Email subject.
- `$message` - Email message.
- `$scheduletime` - Scheduled time for sending the email.
- `$writelog` - Boolean flag indicating whether to write a log.
- `$employee` - Employee ID or object.
- `$campaign` - Campaign ID.
- `$campaign_item` - Campaign item ID.
- `$email_id` - Email ID.
- `$cc` - CC recipients.
- `$bcc` - BCC recipients.

Returns:

- `true` if the email is sent successfully, otherwise `false`.

Tags

`LoadTags($rewritecache)`

Loads the tags associated with the customer.

Parameters:

- `$rewritecache` - Boolean flag indicating whether to rewrite the cache.

`GetTagIDsArray()`

Gets an array of tag IDs associated with the customer.

Returns:

- An array of tag IDs.

`TagsCount()`

Gets the count of tags associated with the customer.

Returns:

- The count of tags.

`HasTagID($tag_id)`

Checks if the customer has a specific tag ID.

Parameters:

- `$tag_id` - Tag ID.

Returns:

- `true` if the customer has the tag, otherwise `false`.

`SetTagID($tag_id)`

Sets a specific tag ID for the customer.

Parameters:

- `$tag_id` - Tag ID.

`UnsetTagID($tag_id)`

Unsets (removes) a specific tag ID associated with the customer.

`GetTagsNamesArray()`

Get an array of tag names associated with the customer.

Parameters: None

Returns: `array`

`SetSelectedTags(array $tag_ids_array)`

Set selected tags for the customer.

Parameters:

- `$tag_ids_array` - Array of tag IDs to be set

Returns: `self`

`UnsetAllTags()`

Unset all tags associated with the customer.

Parameters: None

Returns: None

Attachments

`LoadAttachments($rewritecache)`

Loads the attachments associated with the customer.

Parameters:

- `$rewritecache` - Boolean flag indicating whether to rewrite the cache.

`HasAttachments()`

Checks if the customer has any attachments.

Returns:

- `true` if the customer has attachments, otherwise `false`.

`GetAttachmentsList()`

Gets a list of attachments associated with the customer.

Returns:

- An array containing attachment details.

`UnsetAttachments()`

Unsets (removes) all attachments associated with the customer.

`ClearAttachments()`

Clears (removes and deletes) all email attachment files associated with the customer.

`UnsetAttachmentID($id)`

Unsets (removes) a specific attachment ID associated with the customer.

`AddAttachment($attachment)`

Adds a new attachment to the customer.

Parameters:

- `$attachment` - ID of the attachment to be added.

Referrals

`ReferralID()`

Gets the referral ID associated with the customer.

Returns:

- Referral ID.

`HasReferralID()`

Checks if the customer has a referral ID.

Returns:

- `true` if the customer has a referral ID, otherwise `false`.

`SetReferralID($id_referral)`

Sets the referral ID for the customer.

Parameters:

- `$id_referral` - Referral ID.

`AddedTime()`

Gets the time when the customer was added.

Returns:

- Time of addition.

Phone Information

`HasCellphone()`

Checks if the customer has a main cellphone.

Parameters: None

Returns: Boolean

`HasCellphoneOLD()`

Checks if the customer has a cellphone using an older method.

Parameters: None

Returns: Boolean

`isActiveCellPhone()`

Checks if the customer has an active cellphone.

Parameters: None

Returns: Boolean

`Cellphones(): TPhoneList`

Returns a list of cellphones associated with the customer.

Parameters: None

Returns: TPhoneList

`Cellphone()`

Returns the full phone number of the main cellphone for the customer.

Parameters: None

Returns: Full phone number as a string

`MainCellphoneID()`

Returns the ID of the main cellphone for the customer.

Parameters: None

Returns: Integer

`CellphoneOLD()`

Returns the cellphone number using an older method.

Parameters: None

Returns: Cellphone number as a string

`CellphoneFormatted()`

Returns the formatted main cellphone number with respect to the country.

Parameters: None

Returns: Formatted phone number as a string

`CreateMainCellPhone($val)`

Creates the main cellphone for the customer.

Parameters:

- `val` - Cellphone number.

Returns: No return value.

`SetCellPhone($val)`

Updates the cellphone number of the customer.

Parameters:

- `val` - New cellphone number.

Returns: No return value.

`isActivePhone()`

Checks if the customer's phone is active.

`SetInactivePhone()`

Sets the phone as inactive for the customer.

`SetActivePhone()`

Sets the phone as active for the customer.

Added By Information

`AddedBy()`

Gets the ID of the user who added the customer.

Returns:

- The ID of the

user.

`SetAddedBy($val)`

Sets the ID of the user who added the customer.

Parameters:

- `$val` - User ID.

`HasAddedByEmployeeID()`

Checks if the customer has an added by employee ID.

Returns:

- `true` if the customer has an added by employee ID, otherwise `false`.

`AddedByEmployeeID()`

Gets the added by employee ID associated with the customer.

Returns:

- Added by employee ID.

`SetAddedByEmployeeID($id)`

Sets the added by employee ID for the customer.

Parameters:

- `$id` - Employee ID.

Emails

`UnreadEmailsCount()`

Returns the count of unread emails associated with the customer.

Parameters: None

Returns: Integer

`HasUnsubscribedEmail($email)`

Checks if the given email is present in the 'unsubscribed_emails' table.

Parameters:

- `$email` - Email address to check

Returns: Boolean

`isActiveEmail()`

Checks if the customer has an active email.

Parameters: None

Returns: Boolean

`HasEmail()`

Checks if the customer has an email address.

Parameters: None

Returns: Boolean

`Email()`

Returns the email address of the customer.

Parameters: None

Returns: Email address as a string

`SetEmail($val)`

Updates the email address of the customer.

Parameters:

- `val` - New email address.

Returns: No return value.

`GetEmailStatus()`

Returns the validity status of the customer's email.

Parameters: None

Returns: Email status (integer).

`SetEmailValidated()`

Sets the email of the customer as validated.

Parameters: None

Returns: No return value.

`SetEmailNotValid()`

Sets the email of the customer as not valid.

Parameters: None

Returns: No return value.

`EmailIsActive()`

Checks if the customer's email is active.

`SetInactiveEmail()`

Sets the email as inactive for the customer.

`SetActiveEmail()`

Sets the email as active for the customer.

`GetEmailsCount()`

Returns the count of emails associated with the customer.

`GetLastEmail()`

Get details of the last email sent to the customer.

Parameters: None

Returns: `TCustomerLog` or `null`

`LastEmail()`

Get timestamp of the last email sent/received from the customer.

Parameters: None

Returns: Email timestamp (integer).

`SetLastEmail($val)`

Set timestamp of the last email sent/received from the customer.

- `$val` - Value to set for sent/received email

Returns: none.

Messaging

`SendSMS($text, $messagetype = 'conversation', $scheduletime = 0, $attachments = Array(), $contact_id = 0, $campaign_id = 0, $id_employee = 0)`

Sends an SMS message to the customer. This method only sends a message without writing to the message logs or asking for permissions to receive messages. You should avoid using it if possible and use `SendMessage` instead.

Parameters:

- `text` - Text message to be sent
- `messagetype` - Message type (default: 'conversation')
- `scheduletime` - Scheduled time for the message (default: 0)
- `attachments` - Array of attachments (default: empty array)
- `contact_id` - Contact ID (default: 0)
- `campaign_id` - Campaign ID (default: 0)
- `id_employee` - Employee ID (default: 0)

Returns: Void

`SendInitialSMS()`

Sends an initial SMS message to the customer.

Parameters: None

Returns: Void

`ReSendInitialSMS()`

Re-sends the initial SMS message to the customer.

Parameters: None

Returns: Void

`IncomingSMS($text, $id_employee)`

Handles incoming SMS (Private Business System) messages from customers.

Parameters:

- `text` - Text of the incoming message.
- `id_employee` - Employee ID, defaults to 0.

Returns: No return value.

`isAllowedToReceiveSMSFrom()`

Returns true, indicating that the customer is allowed to receive SMS (Prescription Benefit Services).

Parameters: None

Returns: Boolean

`SendMessageWithAssetImage($text, $asset_id, $logtype = 'conversation', $scheduletime = 0, $send_message_as_sms = true)`

Sends an SMS message with an asset image to the customer.

Parameters:

- `text` - Text message to be sent
- `asset_id` - Asset ID
- `logtype` - Log type (default: 'conversation')
- `scheduletime` - Scheduled time for the message (default: 0)
- `send_message_as_sms` - Boolean, whether to send the message as an SMS (default: true)

Returns: Void

`SendMessage($text, $id_employee = 0, $writelog = true, $logtype = 'conversation', $scheduletime = 0, $attachments = Array(), $send_message_as_sms = true, $asset_id = 0, $contact_id = 0, $campaign_id = 0, $id_campaign_schedule = 0)`

Sends an SMS message to the customer.

Parameters:

- `text` - Text message to be sent
- `id_employee` - Employee ID (default: 0)
- `writelog` - Boolean, whether to write a log (default: true)
- `logtype` - Log type (default: 'conversation')
- `scheduletime` - Scheduled time for the message (default: 0)
- `attachments` - Array of attachments (default: empty array)
- `send_message_as_sms` - Boolean, whether to send the message as an SMS (default: true)
- `asset_id` - Asset ID (default: 0)
- `contact_id` - Contact ID (default: 0)
- `campaign_id` - Campaign ID (default: 0)
- `id_campaign_schedule` - Campaign schedule ID (default: 0)

Returns: Message ID as Integer

`SendVoice($scheduletime = 0, $asset_id = 0)`

Sends a voice message to the customer.

Parameters:

- `scheduletime` - Scheduled time for the message (default: 0)
- `asset_id` - Asset ID (default: 0)

Returns: Void

`GetSMSAcceptedTimestamp()`

Returns the timestamp when SMS messages were accepted by the customer.

Parameters: None

Returns: SMS accepted timestamp (integer).

`SetSMSAcceptedTimestamp($val)`

Sets the SMS accepted timestamp for the customer.

Parameters:

- `val` - New SMS accepted timestamp, defaults to the current time.

Returns: No return value.

`GetSMSAcceptedTag()`

Returns the SMS accepted tag associated with the customer.

Parameters: None

Returns: SMS accepted tag (string).

`SetSMSAcceptedTag($val)`

Sets the SMS accepted tag for the customer.

Parameters:

- `val` - New SMS accepted tag.

Returns: No return value.

`GetSMSPendingTimestamp()`

Returns the timestamp when SMS messages are pending for the customer.

Parameters: None

Returns: SMS pending timestamp (integer).

`SetSMSPendingTimestamp($val)`

Sets the SMS pending timestamp for the customer.

Parameters:

- `val` - New SMS pending timestamp, defaults to the current time.

Returns: No return value.

`isSendingMessagesUndefined()`

Checks if sending SMS messages is in an undefined state for the customer.

Parameters: None

Returns: Boolean value indicating the undefined state.

`isSendingMessagesRejected()`

Checks if sending SMS messages is rejected for the customer.

Parameters: None

Returns: Boolean value indicating the rejection state.

`isSendingMessagesNoResponse()`

Checks if there is no response for sending SMS messages to the customer.

Parameters: None

Returns: Boolean value indicating the no response state.

`isSendingMessagesPending()`

Checks if SMS messages are pending for the customer.

Parameters: None

Returns: Boolean value indicating the pending state.

`isSendingMessagesEnabled()`

Checks if sending SMS messages is enabled for the customer.

Parameters: None

Returns: Boolean value indicating the enabled state.

`isSendingMessagesAbilityExpired()`

Checks if the ability to send SMS messages has expired for the customer.

Parameters: None

Returns: Boolean value indicating the expired state.

`isSendingMessagesFeatureAvaialble()`

Checks if the sending messages feature is available for the customer.

`GetLastSMS()`

Get details of the last SMS (Public Broadcasting Service) sent to the customer.

Parameters: None

Returns: `TCustomerLog` or `null`

`LastText()`

Get timestamp of the last SMS sent/received from the customer.

Parameters: None

Returns: SMS timestamp (integer).

`SetLasttext($val)`

Set timestamp of the last SMS sent/received from the customer.

- `$val` - Value to set for sent/received SMS

Returns: none.

Calls

`SetMissedCall($val)`

Sets the missed call metadata for the customer.

Parameters:

- `$val` - Value to set for missed call

Returns: None

`GetMissedCall()`

Gets the missed call metadata for the customer.

Parameters: None

Returns: Value of missed call metadata

`HasMissedCall()`

Checks if the customer has a missed call.

Parameters: None

Returns: Boolean

`GetCallsCount()`

Returns the count of calls associated with the customer.

`GetBookedCallsCount()`

Returns the count of booked calls associated with the customer.

`GetLastCall()`

Get details of the last call made with the customer.

Parameters: None

Returns: `TCustomerLog` or `null`

`LastCall()`

Get timestamp of the last call made with the customer or received from customer.

Parameters: None

Returns: Call timestamp (integer).

`SetLastCall($val)`

Set timestamp of the last call made with the customer or received from customer.

- `$val` - Value to set for made/received call

Returns: none.

Address

`Address1()`

Returns the first address associated with the customer.

Parameters: None

Returns: Address as a string

`HasAddress1()`

Checks if the customer has the first address.

Parameters: None

Returns: Boolean

`City()`

Returns the city of the customer.

Parameters: None

Returns: City as a string

`HasCity()`

Checks if the customer has a city.

Parameters: None

Returns: Boolean

`State()`

Returns the state of the customer.

Parameters: None

Returns: State as a string

`HasState()`

Checks if the customer has a state.

Parameters: None

Returns: Boolean

`Country()`

Returns the country of the customer.

Parameters: None

Returns: Country as a string

`HasCountry()`

Checks if the customer has a country.

Parameters: None

Returns: Boolean

`ZIP()`

Returns the ZIP code of the customer.

Parameters: None

Returns: ZIP code as a string

`HasZIP()`

Checks if the customer has a ZIP code.

Parameters: None

Returns: Boolean

`AddressFormatted()`

Returns the formatted address of the customer.

Parameters: None

Returns: Formatted address as a string

`SetAddress($val)`

Updates the address of the customer.

Parameters:

- `val` - New address.

Returns: No return value.

`SetCity($val)`

Updates the city of the customer.

Parameters:

- `val` - New city.

Returns: No return value.

`SetState($val)`

Updates the state of the customer.

Parameters:

- `val` - New state.

Returns: No return value.

`SetCountry($val)`

Updates the country of the customer.

Parameters:

- `val` - New country.

Returns: No return value.

`SetZip($val)`

Updates the ZIP code of the customer.

Parameters:

- `val` - New ZIP code.

Returns: No return value.

Name

`Name()`

Returns the full name of the customer.

Parameters: None

Returns: Full name as a string

`HasName()`

Checks if the customer has a name.

Parameters: None

Returns: Boolean

`Initials()`

Returns the initials of the customer based on the first and last names.

Parameters: None

Returns: Initials as a string

`FirstName()`

Returns the first name of the customer.

Parameters: None

Returns: First name as a string

`LastName()`

Returns the last name of the customer.

Parameters: None

Returns: Last name as a string

`ContactName()`

Returns the concatenated first and last names of the customer.

Parameters: None

Returns: Full contact name as a string

`HasNotEmptyContactName()`

Checks if the customer has a non-empty contact name.

Parameters: None

Returns: Boolean

`SetLastName($val)`

Updates the last name of the customer.

Parameters:

- `val` - New last name.

Returns: No return value.

`SetFirstName($val)`

Updates the first name of the customer.

Parameters:

- `val` - New first name.

Returns: No return value.

Registration Information

`isRegisteredByStaff()`

Checks if the customer was registered by staff.

Parameters: None

Returns: Boolean

`isRegisteredOnline()`

Checks if the customer was registered online.

Parameters: None

Returns: Boolean

`isRegisteredViaSMS()`

Checks if the customer was registered via SMS.

Parameters: None

Returns: Boolean

`isRegisteredViaMobileApp()`

Checks if the customer was registered via a mobile app.

Parameters: None

Returns: Boolean

Vehicle Information

`VehicleFormatted()`

Returns the formatted representation of the customer's vehicle details.

Parameters: None

Returns: Formatted vehicle details as a string

`VehicleCondition()`

Returns the vehicle condition associated with the customer.

Parameters: None

Returns: Vehicle condition as a string

`VehicleModel()`

Returns the vehicle model associated with the customer.

Parameters: None

Returns: Vehicle model as a string

`VehicleMake()`

Returns the vehicle make associated with the customer.

Parameters: None

Returns: Vehicle make as a string

URLS

`HasSocialURLS()`

Checks if the customer has any social media URLs.

Parameters: None

Returns: Boolean

`HasFacebookURL()`

Checks if the customer has a Facebook URL.

Parameters: None

Returns: Boolean

`FacebookURL()`

Returns the Facebook URL of the customer.

Parameters: None

Returns: Facebook URL as a string

`HasTwitterURL()`

Checks if the customer has a Twitter URL.

Parameters: None

Returns: Boolean

`TwitterURL()`

Returns the Twitter URL of the customer.

Parameters: None

Returns: Twitter URL as a string

`HasInstagramURL()`

Checks if the customer has an Instagram URL.

Parameters: None

Returns: Boolean

`InstagramURL()`

Returns the Instagram URL of the customer.

Parameters: None

Returns: Instagram URL as a string

`HasLinkedInURL()`

Checks if the customer has a LinkedIn URL.

Parameters: None

Returns: Boolean

`LinkedInURL()`

Returns the LinkedIn URL of the customer.

Parameters: None

Returns: LinkedIn URL as a string

`SetFacebookUrl($val)`

Updates the Facebook URL of the customer.

Parameters:

- `val` - New Facebook URL.

Returns: No return value.

`SetTwitterUrl($val)`

Updates the Twitter URL of the customer.

Parameters:

- `val` - New Twitter URL.

Returns: No return value.

`SetInstagramUrl($val)`

Updates the Instagram URL of the customer.

Parameters:

- `val` - New Instagram URL.

Returns: No return value.

`SetLinkedin($val)`

Updates the LinkedIn profile URL of the customer.

Parameters:

- `val` - New LinkedIn URL.

Returns: No return value.

`DetailsURL()`

Returns the URL for accessing detailed customer information.

Parameters: None

Returns: URL as a string

Sales Data

`SalesPersonID()`

Returns the ID of the salesperson associated with the customer.

Parameters: None

Returns: Integer

`SalesPerson2ID()`

Returns the ID of the secondary salesperson associated with the customer.

Parameters: None

Returns: Integer

`SalesManagerID()`

Returns the ID of the sales manager associated with the customer.

Parameters: None

Returns: Integer

`Value()`

Returns the monetary value associated with the customer.

Parameters: None

Returns: Floating-point number

`GetARR()`

Returns the Annual Recurring Revenue (ARR) based on the customer's value.

Parameters: None

Returns: ARR as a formatted string

`GetMRR()`

Returns the Monthly Recurring Revenue (MRR) based on the customer's value.

Parameters: None

Returns: MRR as a formatted string

`HasValue()`

Checks if the customer has a monetary value.

Parameters: None

Returns: Boolean

`SetValue($value)`

Sets the monetary value for the customer.

Parameters:

- `$value` - Value to set for monetary value

Returns: None

`DealCloseDate()`

Gets the deal close date for the customer.

Parameters: None

Returns: Deal close date as Integer

`HasDealCloseDate()`

Checks if the customer has a deal close date.

Parameters: None

Returns: Boolean

`SetDealCloseDate($value)`

Sets the deal close date for the customer.

Parameters:

- `value` - Deal close date as Integer

Returns: Void

Stage Information

`StageInfoDecoration($status_tag)`

Returns HTML code for visual representation of customer stage information.

Parameters:

- `$status_tag` - Status tag for customer stage

Returns: HTML code as a string

`StageInfo($status_tag)`

Returns an array with the total count of stages and the active stage index.

Parameters:

- `$status_tag` - Status tag for customer stage

Returns: Array with 'total_count' and 'active' keys

`PipelineTypeTitle()`

Returns the title of the pipeline type (e.g., 'Opportunity', 'Customer').

Parameters: None

Returns: Pipeline type title as a string

`PipelineType()`

Returns the pipeline type associated with the customer.

Parameters: None

Returns: Pipeline type as a string

`SetPipelineType($value)`

Updates the pipeline type for the customer.

Parameters:

- `$value` - Value to set for pipeline type

Returns: None

`OpportunityStageOwner()`

Returns the owner information for the opportunity stage.

Parameters: None

Returns: Opportunity stage owner information as a string

`isOpportunityStageOwner()`

Checks if the customer is the owner of the opportunity stage.

Returns: Boolean

`SetOpportunityStageOwner()`

Sets the customer as the owner of the opportunity stage.

Parameters: None

Returns: Void

`UnsetOpportunityStageOwner()`

Unsets the customer as the owner of the opportunity stage.

Parameters: None

Returns: Void

`StageTag()`

Returns the stage tag of the customer.

`HasStageTag()`

Checks if the customer has a stage tag.

`StageName()`

Returns the stage name of the customer.

`StageNamePublic()`

Returns the public stage name of the customer.

`isStageReceived()`

Checks if the customer is in the "received" stage.

`isStageContact()`

Checks if the customer is in the "contact" stage.

`isStageAppointment()`

Checks if the customer is in the "appointment" stage.

`isStageSold()`

Checks if the customer is in the "sold" stage.

`isStageLost()`

Checks if the customer is in the "lost" stage.

`SetStageTag($tagname)`

Sets the stage tag for the customer.

Parameters:

- `$tagname` - Tag name to set for the stage.

Location

`LocationName()`

Returns the name of the location associated with the customer.

Parameters: None

Returns: Location name as a string

`LocationID()`

Gets the location ID associated with the customer.

Parameters: None

Returns: Integer

`HasLocationID()`

Checks if the customer has a location ID.

Parameters: None

Returns: Boolean

`SetLocation($location_or_id)`

Sets the location for the customer.

Parameters:

- `location_or_id` - Location object or ID

Returns: Void

Organization

`Organization()`

Gets the organization title associated with the customer.

Parameters: None

Returns: Organization title as String

`OrganizationID()`

Gets the organization ID associated with the customer.

Parameters: None

Returns: Integer

`HasOrganizationID()`

Checks if the customer has an organization ID.

Parameters: None

Returns: Boolean

`SetOrganizationID($organization_or_id)`

Sets the organization ID for the customer.

Parameters:

- `organization_or_id` - Organization object or ID

Returns: Void

Points

`GetPoints()`

Gets the balance points for the customer.

Parameters: None

Returns: Balance points as Float

`AddOnePoint()`

Adds one point to the balance points of the customer.

Parameters: None

Returns: Void

`wasLeadPaymentMade()`

Checks if a lead payment was made for the customer.

Parameters: None

Returns: Boolean

`wasSuccessLeadPaymentMade()`

Checks if a successful lead payment was made for the customer.

Parameters: None

Returns: Boolean

`CancelLeadPayments()`

Cancels pending lead payments associated with the customer.

Parameters: None

Returns: Void

Conversations

`MarkAsConversationUnread()`

Marks the conversation as unread for the customer.

Parameters: None

Returns: Void

`MarkAsConversationRead()`

Marks the conversation as read for the customer.

Parameters: None

Returns: Void

`HasUnreadConversation()`

Checks if the customer has unread conversations.

Parameters: None

Returns: Boolean

Contact List

`LoadContactList($rewritecache = false)`

Loads the contact list for the customer.

Parameters:

- `rewritecache` - Boolean, whether to rewrite the cache

Returns: Void

`GetContactListIDsArray()`

Gets an array of contact list IDs for the customer.

Parameters: None

Returns: Array of contact list IDs

`LoadLists($rewritecache)`

Loads the lists associated with the customer.

Parameters:

- `$rewritecache` - Boolean flag indicating whether to rewrite the cache.

`UnsetContactIDs()`

Unsets (removes) all contact IDs associated with the customer.

Member Type

`isMemberTypeReferral()`

Checks if the customer is of member type 'referral'.

Parameters: None

Returns: Boolean

`MemberType()`

Gets the member type of the customer.

Parameters: None

Returns: Member type as String

`SetMemberTypeReferral()`

Sets the member type of the customer to "referral".

Parameters: None

Returns: No return value.

Financial Information

`SetBankName($val)`

Updates the bank name associated with the customer.

Parameters:

- `val` - New bank name.

Returns: No return value.

`SetRouting($val)`

Updates the routing information associated with the customer.

Parameters:

- `val` - New routing information.

Returns: No return value.

`SetAcctNumber($val)`

Updates the account number associated with the customer.

Parameters:

- `val` - New account number.

Returns: No return value.

`SetPaypalEmail($val)`

Updates the PayPal email associated with the customer.

Parameters:

- `val` - New PayPal email.

Returns: No return value.

`GetBankName()`

Returns the bank name associated with the customer.

Parameters: None

Returns: Bank name (string).

`GetRouting()`

Returns the routing information associated with the customer.

Parameters: None

Returns: Routing information (string).

`GetSetAcctNumber()`

Returns the account number associated with the customer.

Parameters: None

Returns: Account number (string).

`GetSetPaypalEmail()`

Returns the PayPal email associated with the customer.

Parameters: None

Returns: PayPal email (string).

Appointments

`HasAppointments()`

Checks if the customer has appointments.

Parameters: None

Returns: Boolean value indicating the presence of appointments.

`MarkAsHasAppointments()`

Marks the customer as having appointments.

Parameters: None

Returns: No return value.

User Information

`UserID()`

Returns the user ID associated with the customer.

Parameters: None

Returns: User ID (integer).

`SetUserID($val)`

Sets the user ID for the customer.

Parameters:

- `val` - New user ID.

Returns: No return value.

Campaigns

`AddToCampaign($campaign)`

Adds the customer to a marketing campaign.

Parameters:

- `campaign` - Campaign to add the customer to.

Returns: No return value.

`RemoveFromCampaign($campaign)`

Removes the customer from a marketing campaign.

Parameters:

- `campaign` - Campaign to remove the customer from.

Returns: No return value.

`GetCampaignsCount()`

Returns the count of campaigns associated with the customer.

`LoadCampaigns($rewritecache)`

Loads the campaigns associated with the customer.

Parameters:

- `$rewritecache` - Boolean flag indicating whether to rewrite the cache.

`UnsetCampaigns()`

Unsets (removes) all campaigns associated with the customer.

`GetCampaignsList()`

Gets a list of campaigns associated with the customer.

Returns:

- An array containing campaign details.

`CampaignID()`

Returns the campaign ID associated with the customer.

`SetCampaignID($id)`

Sets the campaign ID for the customer.

`SetCampaignTypeID($id)`

Sets the campaign type ID for the customer.

`HasCampaignID()`

Check if the customer has a campaign ID.

Parameters: None

Returns: `bool`

`StartCampaignAction($id_object, $scheduledtime)`

Log the start of a campaign for the customer.

Parameters:

- `$id_object` - ID of the campaign
- `$scheduledtime` - Scheduled time for the campaign start

Returns: None

`StopCampaignAction($id_object, $scheduledtime)`

Log the stop of a campaign for the customer.

Parameters:

- `$id_object` - ID of the campaign
- `$scheduledtime` - Scheduled time for the campaign stop

Returns: None

`LoadSequences($rewritecache)`

Loads the sequences associated with the customer.

Parameters:

- `$rewritecache` - Boolean flag indicating whether to rewrite the cache.

`UnsetSequences()`

Unsets (removes) all sequences associated with the customer.

`GetActiveSequence()`

Get the active campaign sequence for the customer.

Parameters: None

Returns: `array` or `null`

`GetCustomerSequence()`

Gets the customer's sequence.

Returns: The ID of the customer's sequence.

`SetUnsubscribeStatus($val)`

Sets the unsubscribe status for the customer.

Parameters:

- `val` - Unsubscribe status.

Returns: No return value.

`GetUnsubscribeStatus()`

Returns the unsubscribe status of the customer.

`isUnsubscribeStatus()`

Checks if the customer has an unsubscribe status.

`SetUnsubscribeFromTextStatus($val)`

Sets the unsubscribe from text (SMS) status for the customer.

Parameters:

- `$val` - Value to set for unsubscribe from text status.

`GetUnsubscribeFromTextStatus()`

Returns the unsubscribe from text (SMS) status of the customer.

`isUnsubscribeFromTextStatus()`

Checks if the customer has an unsubscribe from text (SMS) status.

`StopCampaign($campaign)`

Stop a campaign for the customer.

Parameters:

- `$campaign` - ID of the campaign to be stopped

Returns: `bool`

`SwitchToCampaign($new_campaign)`

Switch the customer to a new campaign.

Parameters:

- `$new_campaign` - ID of the new campaign

Returns: `bool`

Taxonomy

`GetTaxonomy($object_name, $object_id, $use_object_id_as_rel_id=false)`

Gets taxonomy information for the customer.

Parameters:

- `$object_name` - Object name for taxonomy.
- `$object_id` - Object ID for taxonomy.
- `$use_object_id_as_rel_id` - Flag to use object ID as rel ID.

Returns: An array of taxonomy IDs.

`SetTaxonomy($object_name, $object_id, $rel_id)`

Sets taxonomy information for the customer.

Parameters:

- `$object_name` - Object name for taxonomy.
- `$object_id` - Object ID for taxonomy.
- `$rel_id` - Relationship ID for taxonomy.

`UnsetTaxonomy($object_name, $object_id, $rel_id)`

Unsets taxonomy information for the customer.

Parameters:

- `$object_name` - Object name for taxonomy.
- `$object_id` - Object ID for taxonomy.
- `$rel_id` - Relationship ID for taxonomy.

Metadata

`SetMetaData($key, $val)`

Sets metadata for the customer.

Parameters:

- `key` - Metadata key.
- `val` - Metadata value.

Returns: No return value.

`GetMetaData($key)`

Gets metadata for the customer.

Parameters:

- `key` - Metadata key.

Returns: Metadata value.

Actions

`AppointmentAction($id_object, $scheduledtime)`

Log an appointment action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$scheduledtime` - Scheduled time for the appointment

Returns: None

`SendSMSAction($id_object, $scheduledtime, $bulk, $id_employee)`

Log a SMS (Public Broadcasting Service) action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$scheduledtime` - Scheduled time for the action
- `$bulk` - Boolean indicating if it's a bulk action
- `$id_employee` - ID of the employee performing the action

Returns: None

`IncomingSMSAction($id_object, $id_employee)`

Log an incoming SMS action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee receiving the SMS

Returns: None

`CreateAppointmentAction($id_object, $id_employee)`

Log the creation of an appointment action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee creating the appointment

Returns: None

`CreateBookedCallAction($id_object, $id_employee)`

Log the creation of a booked call action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee creating the booked call

Returns: None

`CreateNoteAction($id_object, $id_employee)`

Log the creation of a note action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee creating the note

Returns: None

`SendEmailAction($id_object, $scheduledtime, $bulk, $id_employee)`

Log an email action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$scheduledtime` - Scheduled time for the email
- `$bulk` - Boolean indicating if it's a bulk email
- `$id_employee` - ID of the employee sending the email

Returns: None

`IncomingEmailAction($id_object, $id_employee)`

Log an incoming email action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee receiving the email

Returns: None

`CallAction($id_object, $id_employee)`

Log a call action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee making the call

Returns: None

`IncomingCallAction($id_object, $id_employee)`

Log an incoming call action for the customer.

Parameters:

- `$id_object` - ID of the related object
- `$id_employee` - ID of the employee receiving the call

Returns: None

Confidence Level

`SetConfidenceLevel($val)`

Sets the confidence level for the customer.

Parameters:

- `$val` - Value to set for confidence level.

`HasConfidenceLevel()`

Checks if the customer has a confidence level.

`GetConfidenceLevel()`

Gets the confidence level of the customer.

Counts

`GetActivitiesCount()`

Returns the count of activities associated with the customer.

`GetTextMessagesCount()`

Returns the count of text messages associated with the customer.

`GetTasksCount()`

Returns the count of tasks associated with the customer.

`GetMarketingMessagesCount()`

Returns the count of marketing messages associated with the customer.

`MarketingMessagesCount()`

Returns the count of marketing messages for the customer.

Parameters: None

Returns: Count of marketing messages (integer).

`UnreadMessagesCount()`

Returns the count of unread messages for the customer.

Parameters: None

Returns: Count of unread messages (integer).

`InteractionsCount()`

Returns the total count of interactions with the customer.

Parameters: None

Returns: Integer

Sort Order

`SortOrder()`

Returns the sort order of the customer.

`SetSortOrder($val)`

Sets the sort order for the customer.

Parameters:

- `$val` - Value to set for sort order.

Role

`SetRole($val)`

Sets the role for the customer.

Parameters:

- `$val` - Value to set for the role.

`Role()`

Returns the role of the customer.

`HasRole()`

Checks if the customer has a role.

Source

`SourcePlatformName()`

Returns the source platform name for the customer.

`HasSourcePlatformName()`

Checks if the customer has a source platform name.

`SetSourcePlatformName($name)`

Sets the source platform name for the customer.

Parameters:

- `$name` - Name to set for the source platform.

`Source()`

Returns the source information for the customer.

`SetSource($name)`

Sets the source information for the customer.

Parameters:

- `$name` - Name to set for the source.

`isSourceOffer()`

Check if the customer source is 'offer'.

Parameters: None

Returns: `bool`

Approvement

`isApproved()`

Checks if the customer is approved.

Parameters: None

Returns: Boolean

`SetCustomerApproved()`

Sets the customer as approved.

Parameters: None

Returns: Void

`SetCustomerNotApproved()`

Sets the customer as not approved.

Parameters: None

Returns: Void

`SetStatusApproved()`

Sets the customer status as approved and performs additional actions.

Parameters: None

Returns: Void

Profile Photo

`HasProfilePhoto()`

Checks if the customer has a profile photo.

Parameters: None

Returns: Boolean value indicating the presence of a profile photo.

`ProfilePhotoPath()`

Returns the file path for the customer's profile photo.

Parameters: None

Returns: File path (string).

`ProfilePhotoURL()`

Returns the URL of the customer's profile photo.

Parameters: None

Returns: URL (string).

Miscellaneous

`Touches()`

Returns the touches count for the customer.

`SetTouches($val)`

Sets the touches count for the customer.

Parameters:

- `$val` - Value to set for touches count.

`vCardURL($full_url=false)`

Generates the vCard URL for the customer.

Parameters:

- `$full_url` - Flag to include the full URL.

Returns: The vCard URL.

`SendVCard()`

Sends the vCard for the customer.

`ContentID()`

Returns the content ID associated with the customer.

`HasContentID()`

Checks if the customer has a content ID.

`SetContentID($name)`

Sets the content ID for the customer.

Parameters:

- `$name` - Name to set for the content ID.

`SetSourcePlatformNameBySymbol($symbol)`

Sets the source platform name based on a symbol.

Parameters:

- `$symbol` - Symbol to set the source platform name.

`GetNextActivity()`

Get the next scheduled activity for the customer.

Parameters: None

Returns: `array` or `null`

`GetLastContact()`

Get the details of the last contact made with the customer.

Parameters: None

Returns: `array` or `null`

`GetLastNote()`

Get details of the last note created for the customer.

Parameters: None

Returns: `TCustomerNote` or `null`

`Exists()`

Checks if the customer exists.

Returns:

- `true` if the customer exists, otherwise `false`.

`isDeleted()`

Checks if the customer is marked as deleted.

Parameters: None

Returns: Boolean

`Disable()`

Disables the customer.

`isEnabled()`

Checks if the customer is enabled.

Returns:

- `true` if the customer is enabled, otherwise `false`.

`Enable()`

Enables the customer by updating the value of 'is_enabled' to 1.

Parameters: None

Returns: None

`SetRegisteredType($val)`

Updates the registered type of the customer.

Parameters:

- `$val` - Value to set for registered type

Returns: None

`InactiveDays()`

Returns the number of days since the last contact with the customer.

Parameters: None

Returns: Integer

`LastContactedData()`

Returns the date of the last contact with the customer in a formatted manner.

Parameters: None

Returns: Formatted date as a string

`Note($nl2br = false)`

Returns the note associated with the customer, optionally converting newlines to HTML line breaks.

Parameters:

- `$nl2br` - Boolean indicating whether to convert newlines to HTML line breaks

Returns: Note as a string

`SetNote($val)`

Sets the note for the customer after sanitizing and removing HTML tags.

Parameters:

- `$val` - Value to set for the note

Returns: None

`ID()`

Returns the ID of the customer.

Parameters: None

Returns: Integer

`Remove()`

Removes associated records and deletes the customer.

Parameters: None

Returns: Void

`Log($text)`

Logs a message for the customer.

Parameters:

- `text` - Text message to be logged

Returns: Void

`UnsetStoredPassword()`

Unsets the stored password for the customer.

Parameters: None

Returns: Void

`SetLastUpdateTime()`

Updates the last update time for the customer.

Parameters: None

Returns: None

`CompanyID()`

Returns the company ID associated with the customer.

Parameters: None

Returns: Company ID (integer).

`UpdateValues($params)`

Updates customer information with the provided parameters.

Parameters:

- `params` - Associative array of key-value pairs representing customer information.

Returns: No return value.

CustomerCellPhone Class

- Full class name: `\GS\Customer\Decorator\CustomerDecoratorPrototype`

Information:

- Type: Decorator
- Decoration for class: \GS\Customer\Customer
- Extends: \GS\Customer\Decorator\CustomerDecoratorPrototype

`GetCustomerCellPhonesList()`

Gets a list of cell phones associated with the customer.

Returns: `TPhoneList` - List of cell phones.

`GetCustomerCellPhones()`

Gets an array of formatted customer cell phones.

Returns: `array` - An array containing phone details such as prefix and short phone.

`GetAdditionalPhones()`

Gets additional phones associated with the customer, excluding the main phone.

Returns: `TPhoneList` - List of additional phones.

`GetAdditionalPhonesArray()`

Gets an array of formatted additional phones associated with the customer.

Returns: `array` - An array containing phone details including ID, type, formatted phone, and active status.

`GetAllPhonesArrayWithFlag()`

Gets an array of all customer cell phones with various details.

Returns: `array` - An array containing phone details including type, formatted phone, prefix, short phone, country, and extension.

`isDublicateWithAnotherCustomer($prefix, $phone)`

Checks if there is a duplicate phone with the given prefix and phone number associated with another customer.

Parameters:

- `$prefix` - `string` - The country prefix.
- `$phone` - `string` - The phone number.

Returns: `bool` - Returns `true` if there is a duplicate, otherwise `false`.

`isDublicate($prefix, $phone, $company = null)`

Checks if there is a duplicate phone with the given prefix and phone number in the specified company or the current company.

Parameters:

- `$prefix` - `string` - The country prefix.
- `$phone` - `string` - The phone number.
- `$company` - `TCompany` (optional) - The company to check for duplicates. Defaults to the current company.

Returns: `bool` - Returns `true` if there is a duplicate, otherwise `false`.

`isCustomersPhone($prefix, $phone)`

Checks if the provided phone with the given prefix is associated with the current customer.

Parameters:

- `$prefix` - `string` - The country prefix.
- `$phone` - `string` - The phone number.

Returns: `bool` - Returns `true` if the phone is associated with the current customer, otherwise `false`.

AddCustomerPhone($phone_prefix, $short_phone, $country, $inactive_phone = false, $phone_type = TPhone::OTHER_PHONE, $extension='')

Adds a phone to the existing customer.

Parameters:

- `$phone_prefix` - `string` - The country prefix.
- `$short_phone` - `string` - The short phone number.
- `$country` - `string` - The country of the phone number.
- `$inactive_phone` - `bool` (optional) - Flag indicating if the phone is inactive. Defaults to `false`.
- `$phone_type` - `string` (optional) - The type of the phone. Defaults to `TPhone::OTHER_PHONE` .
- `$extension` - `string` (optional) - The phone extension. Defaults to an empty string.

Returns: `void`

RemoveCustomerPhones()

Removes all phones associated with the current customer.

Returns: `void`

HasAdditionalPhones()

Checks if the customer has additional phones (excluding the main phone).

Returns: `bool` - Returns `true` if there are additional phones, otherwise `false`.

MainCellphone()

Gets the main cellphone associated with the customer.

Returns: `TPhone` - The main cellphone, or a placeholder `TPhone` object if not found.

HasMainCellphone()

Checks if the customer has a main cellphone.

Returns: `bool` - Returns `true` if there is a main cellphone, otherwise `false`.

CustomerDecoratorPrototype class

- Full class name: `\GS\Customer\Decorator\CustomerDecoratorPrototype`

Information:

- Type: Decorator Prototype
- Decoration for class: \GS\Customer\Customer

`\GS\Customer\Decorator\CustomerDecoratorPrototype` is designed to facilitate the separation of customer-related tasks for developers. You should extend this class to add several functions for a specific purpose. For example, a class that implements all aspects of working with a customer profile can be built in a single decorator and then executed from it. The decorator class takes an already initiated instance of `\GS\Customer\Customer` as a constructor parameter. All updates are made on the provided instance, immediately affecting it.

`__construct($customer)`

Constructor for `CustomerDecoratorPrototype`.

Parameters:

- `$customer` - `Customer` instance to be associated with the decorator.

`Customer()`

Parameters: None

Returns: `Customer` - The associated customer instance.

`Init($customer)`

Static method to initialize and create a new instance of `CustomerDecoratorPrototype`.

Parameters:

- `$customer` - `Customer` instance to be associated with the decorator.

Returns: `CustomerDecoratorPrototype` - A new instance of the decorator initialized with the provided customer.

Customer List class

- Full class name: `\GS\Customer\CustomerList`
- Legacy alias (shouldn't be used in a new code): ~~`\TCustomerList`~~

Information:

- Type: ORM Model List
- DB Table: `customers`
- List Items: \GS\Customer\Customer

Table of Contents

Loading

Loading selected records into memory helps to work with them in a fast and efficient way. Loaded records can be processed multiple times, and you will have convenient functions for extracting and updating the data.

However, it is not a good approach for handling a large number of records as it may slow down the server. For dealing with a substantial amount of records, use the `Open` and `Fetch` methods to retrieve and process each record individually.

`Load($sql=''): void`

Loads customer data from the database based on the generated SQL query.

Parameters:

- `sql` - Additional SQL conditions to be added to the query.

Returns: `void`

`Item(int $index): Customer`

Returns a customer object at the specified index in the items array.

Parameters:

- `index` - Index of the customer in the items array.

Returns: `Customer` - The customer object.

`EachItem()`

This function utilizes a generator to iterate over each item in the collection.

Parameters: None

Returns: A generator that yields each item in the collection.

~~`EachItemOfLoaded()`~~

A legacy alias of `EachItem()`. Shouldn't be used in a new code.

`Count(): int`

Gets the count of customers in the items array.

Returns: `int` - Count of customers.

`ExtractIDsArray(): array`

Extracts and returns an array of loaded customer IDs as an array.

Parameters: None

Returns: `array` - An array of customer IDs.

`ExtractOrganizationsIDsArray(): array`

Extracts and returns an array of organization IDs from the loaded items as an array.

Parameters: None

Returns: `array` - An array of organization IDs.

`ExtractNamesArray($addrolename=false)`

Extracts an array of names from the customer objects in the class. Optionally, includes the primary role title in parentheses if the `$addrolename` parameter is set to true.

Parameters:

- `$addrolename` - Boolean flag indicating whether to include the primary role name. Default is `false`.

Returns: Array

Opening

Working with a large number of records without loading them into memory can be accomplished using the `Open` and `Fetch` methods.

`Open(): self`

Executes the SQL query and fetches the results.

Parameters: None

Returns: `self`

`Fetch(): TCustomer|false`

Fetches the next customer from the query result.

Parameters: None

Returns: `TCustomer|false` - The next customer object or false if there are no more results.

Limit and offset

`Limit($limit)`

Sets the limit for the number of records to be retrieved.

Parameters:

- `limit` - The limit for the number of records.

Returns: `void`

`Offset($offset)`

Sets the offset for the records to be retrieved.

Parameters:

- `offset` - The offset for the records.

Returns: `void`

`GetCurrentLimit()`

Gets the current limit value.

Returns: int

`GetCurrentOffset()`

Gets the current offset value.

Returns: int

Aggregation

`TotalCount(bool $reloadcache=false): int`

Gets the total count of customers based on the current filter criteria.

Parameters:

- `reloadcache` - Flag to force reloading the cache.

Returns: `int` - Total count of customers.

`TotalValue($rewritecache=false): float`

Gets the total value of customers based on the current filter criteria.

Parameters:

- `rewritecache` - Flag to force rewriting the cache.

Returns: `float` - Total value of customers.

Ordering

`OrderBySMSAcceptPendingTimestamp($asc=true)`

Sets the order of the query results based on SMS acceptance pending timestamp.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByUnreadOrName($asc=true)`

Sets the order of the query results based on unread count or customer names.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByUnreadOrLastUpdate($asc=true)`

Sets the order of the query results based on unread count or last update timestamp.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByMultipleField($orderByStr)`

Sets the order of the query results based on a custom order-by string.

Parameters:

- `orderByStr` - Custom order-by string.

Returns: `void`

`OrderByID($asc=true)`

Sets the order of the query results based on customer IDs.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByValues($asc=true)`

Sets the order of the query results based on customer values.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByName($asc=true)`

Sets the order of the query results based on customer names.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByConfidenceLevel($asc=true)`

Sets the order of the query results based on confidence levels.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByDealCloseDate($asc=true)`

Sets the order of the query results based on deal close date.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderByTouches($asc=true)`

Sets the order of the query results based on the number of touches.

Parameters:

- `asc` - Flag to indicate ascending or descending order.

Returns: `void`

`OrderBySortOrder()`

Sets the order of the query results based on sort order and last update timestamp.

Parameters: None

Returns: `void`

Filtering

`SetFilterNotDeletedOr($ids)`

Sets the filter to include non-deleted customers or customers with specified IDs.

Parameters:

- `ids` - An array of customer IDs.

Returns: `void`

`SetFilterStage($stage)`

Sets the filter for customers in a specific stage.

Parameters:

- `stage` - The target stage.

Returns: `void`

`SetFilterEmailBeginnig(string $beginning): void`

Sets the filter to include customers with an email beginning with the specified string.

Parameters:

- `beginning` - The beginning of the email.

Returns: `void`

`SetFilterApproved()`

Sets the filter to include approved customers.

Parameters: None

Returns: `void`

`SetFilterNotApproved()`

Sets the filter to include not approved customers.

Parameters: None

Returns: `void`

`SetFilterReferral($referral)`

Sets the filter to include customers with a specific referral.

Parameters:

- `referral` - Referral information.

Returns: `void`

`SetFilterDay()`

Sets the filter to include customers added on the current day.

Parameters: None

Returns: `void`

`SetFilterWeek()`

Sets the filter to include customers added within the current week.

Parameters: None

Returns: `void`

`SetFilter7days()`

Sets the filter to include customers added within the last 7 days.

Parameters: None

Returns: `void`

`SetFilter14days()`

Sets the filter to include customers added within the last 14 days.

Parameters: None

Returns: `void`

`SetFilter30days()`

Sets the filter to include customers added within the last 30 days.

Parameters: None

Returns: `void`

`SetFilterMonth()`

Sets the filter to include customers added within the current month.

Parameters: None

Returns: `void`

`SetFilter10DaysOld()`

Sets the filter to include customers updated within the last 10 days.

Parameters: None

Returns: `void`

`SetFilterTouches($val)`

Sets the filter to include customers with a specific number of touches.

Parameters:

- `val` - Number of touches.

Returns: `void`

`SetFilterExcludeEmptyDealCloseDate()`

Sets the filter to exclude customers with an empty deal close date.

Parameters: None

Returns: `void`

`SetFilterIDs($array)`

Sets the filter to include customers with specific IDs.

Parameters:

- `array` - An array of customer IDs.

Returns: `void`

`SetFilterReferralIDs($array)`

Sets the filter to include customers with specific referral IDs.

Parameters:

- `array` - An array of referral IDs.

Returns: `void`

`SetFilterPipelineTag($status_tag)`

Sets the filter to include customers with a specific pipeline tag.

Parameters:

- `status_tag` - Pipeline status tag.

Returns: `void`

`SetFilterCustomersPipeline()`

Sets the filter to include customers in the "Customer" pipeline.

Parameters: None

Returns: `void`

`SetFilterOpportunitiesPipeline()`

Sets the filter to include customers in the "Opportunity" pipeline.

Parameters: None

Returns: `void`

`SetFilterOpportunityStageOwner($stage)`

Sets the filter

to include opportunities with a specific stage owner.

Parameters:

- `stage` - Opportunity stage owner.

Returns: `void`

`SetFilterLeadsPipeline()`

Sets the filter to include customers in the "Lead" pipeline.

Parameters: None

Returns: `void`

`SetFilterCompany($id_company)`

Sets the filter to include customers associated with a specific company.

Parameters:

- `id_company` - ID of the company.

Returns: `void`

`SetFilterCompaniesIDs($array)`

Sets the filter to include customers associated with specific company IDs.

Parameters:

- `array` - An array of company IDs.

Returns: `void`

`SetFilterFirstNameBeginnig(string $beginning): void`

Sets the filter to include customers with a first name beginning with the specified string.

Parameters:

- `beginning` - The beginning of the first name.

Returns: `void`

`SetFilterNameLike(string $like_condition): void`

Sets the filter to include customers with names or email matching the specified condition.

Parameters:

- `like_condition` - The condition to match names or email.

Returns: `void`

`SetFilterCustomQuery($query)`

Sets the filter based on a custom SQL query.

Parameters:

- `query` - Custom SQL query.

Returns: `void`

`SetFilterContent($id_content)`

Sets the filter to include customers associated with specific content.

Parameters:

- `id_content` - ID of the content.

Returns: `void`

`SetFilterEnabled()`

Sets the filter to include enabled customers.

Parameters: None

Returns: `void`

`SetFilterDisabled()`

Sets the filter to include disabled customers.

Parameters: None

Returns: `void`

`SetFilterContacts()`

Sets the filter to include customers with a member type of "contact."

Parameters: None

Returns: `void`

`SetFilterMemberTypeReferral()`

Sets the filter to include customers with a member type of "referral."

Parameters: None

Returns: `void`

`SetFilterNotDeleted()`

Sets the filter to include non-deleted customers.

Parameters: None

Returns: `void`

`SetFilterUnassignedPartner()`

Sets the filter to include customers added by unassigned partners.

Parameters: None

Returns: `void`

`SetFilterAddedBy($employee_or_id)`

Sets the filter to include customers added by a specific employee or ID.

Parameters:

- `employee_or_id` - Employee or ID.

Returns: `void`

`SetFilterOffer($id_campaign)`

Sets the filter to include customers associated with a specific campaign.

Parameters:

- `id_campaign` - ID of the campaign.

Returns: `void`

`SetFilterOrganization($id_organization)`

Sets the filter to include customers associated with a specific organization.

Parameters:

- `id_organization` - ID of the organization.

Returns: `void`

`SetFilterSMSAcceptedTag($tag)`

Sets the filter to include customers with a specific SMS accepted tag.

Parameters:

- `tag` - SMS accepted tag.

Returns: `void`

`SetFilterCellPhone($val)`

Sets the filter to include customers with a specific cell phone number.

Parameters:

- `val` - Cell phone number.

Returns: `void`

`SetFilterCellPhoneLike(string $like_condition): void`

Sets the filter to include customers with cell phone numbers matching the specified condition.

Parameters:

- `like_condition` - The condition to match cell phone numbers.

Returns: `void`

`SetFilterRegisteredFrom($val)`

Sets the filter to include customers registered from a specific timestamp.

Parameters:

- `val` - Timestamp.

Returns: `void`

`SetFilterUnread()`

Sets the filter to include unread customers.

Parameters: None

Returns: `void`

`SetFilterRegisteredTo($val)`

Sets the filter to include customers registered up to a specific timestamp.

Parameters:

- `val` - Timestamp.

Returns: `void`

`SetFilterLocation($val)`

Sets the filter to include customers associated with a specific location.

Parameters:

- `val` - ID of the location.

Returns: `void`

`SetFilterName($val)`

Sets the filter to include customers with names containing the specified string.

Parameters:

- `val` - The string to match in names.

Returns: `void`

`SetFilterNameOrEmail($val)`

Sets the filter to include customers with names or email containing the specified string.

Parameters:

- `val` - The string to match in names or email.

Returns: `void`

`SetFilterVehicleMake($val)`

Sets the filter to include customers with a specific vehicle make.

Parameters:

- `val` - Vehicle make.

Returns: `void`

`SetFilterVehicleModel($val)`

Sets the filter to include customers with a specific vehicle model.

Parameters:

- `val` - Vehicle model.

Returns: `void`

`SetFilterVehicleCondition($val)`

Sets the filter to include customers with a specific vehicle condition.

Parameters:

- `val` - Vehicle condition.

Returns: `void`

`SetFilterSMSAcceptPendingTimeLTE($time=NULL)`

Sets the filter to include customers with SMS pending time less than or equal to the specified time.

Parameters:

- `time` - Timestamp.

Returns: `void`

`SetFilterSMSAcceptPending1()`

Sets the filter to include customers with an SMS acceptance tag of 'pending1'.

Parameters: None

Returns: `void`

`SetFilterSMSAcceptPending2()`

Sets the filter to include customers with an SMS acceptance tag of 'pending2'.

Parameters: None

Returns: `void`

`SetFilterSMSAcceptNoResponse()`

Sets the filter to include customers with an SMS acceptance tag of 'noresponse'.

Parameters: None

Returns: `void`

`SetFilterHasAppointments()`

Sets the filter to include customers with appointments.

Parameters: None

Returns: `void`

`SetFilterConfidenceLevel($value)`

Sets the filter to include customers with a specific confidence level.

Parameters:

- `value` - Confidence level.

Returns: `void`

`ShowDeleted()`

Sets the flag to show deleted customers in the results.

Parameters: None

Returns: `void`

`SetFilterExcludeIDs($arrayids)`

Sets the filter to exclude customers with specific IDs.

Parameters:

- `arrayids` - An array of customer IDs.

Returns: `void`

`OrderByLastUpdate($asc=true)`

Sets the sorting order of the customer objects based on their last update time.

Parameters:

- `$asc` - Boolean flag indicating the sorting order. If `true`, ascending order is applied; if `false`, descending order is applied.

Returns: $this

`SetFilterCustomPeriod(int $start_period, int $end_period)`

Sets a custom time period filter for customer objects based on the time they were added.

Parameters:

- `$start_period` - Start timestamp of the custom period.
- `$end_period` - End timestamp of the custom period.

Returns: $this

`SetFilterFirstName(string $first_name)`

Sets a filter for customer objects based on their first name.

Parameters:

- `$first_name` - The first name to filter on.

Returns: $this

`SetFilterLastName(string $last_name)`

Sets a filter for customer objects based on their last name.

Parameters:

- `$last_name` - The last name to filter on.

Returns: $this

GenericSMS class

- Full class name: `\GS\Messages\SMS\GenericSMS`
- Legacy alias (shouldn't be used in a new code): ~~`\TGenericMessage`~~

Information:

- Type: abstract ORM Model
- Descendants: `\GS\Messages\SMS\SMSMessage`, `\GS\Messages\SMS\PendingSMSMessage`
- DB Table: `smslog`, `smslog_pending`

This class is designed to work with SMS conversations. You shouldn't use this class directly.

Table of Contents

General Methods

`ID()`

Returns the integer representation of the SMS message ID.

Returns: Integer - The SMS message ID.

`Exists()`

Checks if the SMS message has a valid ID.

Returns: Boolean - True if the SMS message has a valid ID, false otherwise.

`UpdateValues($params)`

Updates multiple values of the SMS message.

Parameters:

- `$params` - Array - The associative array of parameters to update.

`Remove()`

Removes the SMS message from the database.

Linked Objects

`CompanyID()`

Returns the integer representation of the company ID associated with the SMS message.

Returns: Integer - The company ID.

`SetCompanyID($val)`

Updates the company ID associated with the SMS message.

Parameters:

- `$val` - Integer - The new company ID.

`CustomerID()`

Returns the integer representation of the customer ID associated with the SMS message.

Returns: Integer - The customer ID.

`SetCustomerID($val)`

Updates the customer ID associated with the SMS message.

Parameters:

- `$val` - Integer - The new customer ID.

`OrganizationID()`

Returns the integer representation of the organization ID associated with the SMS message.

Returns: Integer - The organization ID.

`SetOrganizationID($val)`

Updates the organization ID associated with the SMS message.

Parameters:

- `$val` - Integer - The new organization ID.

`ReferralID()`

Returns the referral ID associated with the SMS message.

Returns: Referral ID.

`SetReferralID($val)`

Updates the referral ID associated with the SMS message.

Parameters:

- `$val` - Referral ID.

`EmployeeID()`

Returns the integer representation of the employee ID associated with the SMS message.

Returns: Integer - The employee ID.

`SetEmployeeID($val)`

Updates the employee ID associated with the SMS message.

Parameters:

- `$val` - Integer - The new employee ID.

`CampaignID()`

Returns the integer representation of the campaign ID associated with the SMS message.

Returns: Integer - The campaign ID.

`ContactID()`

Returns the integer representation of the contact ID associated with the SMS message.

Returns: Integer - The contact ID.

`CampaignScheduleID()`

Returns the integer representation of the campaign schedule ID associated with the SMS message.

Returns: Integer - The campaign schedule ID.

Message Information

`TypeTag()`

Returns the type tag associated with the SMS message.

Returns: Type tag.

`SetTypeTag($val)`

Updates the type tag associated with the SMS message.

Parameters:

- `$val` - Type tag.

`isIncoming()`

Checks if the SMS message is incoming.

Returns: Boolean - True if the SMS message is incoming, false otherwise.

`SetIncoming($val = true)`

Sets the SMS message as incoming.

Parameters:

- `$val` - Boolean - Optional. Set to `true` to mark the message as incoming and `false` as outgoing.

`UnsetIncoming()`

Unsets the incoming status of the SMS message and marks it as outgoing.

`Cellphone()`

Returns the cellphone associated with the SMS message.

Returns: Cellphone.

`SetCellphone($val)`

Updates the cellphone associated with the SMS message.

Parameters:

- `$val` - Cellphone.

`HasCellphone()`

Checks if the SMS message has a valid cellphone.

Returns: Boolean - True if the SMS message has a valid cellphone, false otherwise.

`Text()`

Returns the text content of the SMS message.

Returns: Text content.

`SetText($val)`

Updates the text content of the SMS message.

Parameters:

- `$val` - Text content.

Timestamps

`AddedTimestamp()`

Returns the integer representation of the timestamp when the SMS message was added.

Returns: Integer - The timestamp.

`SetAddedTimestamp($val)`

Updates the timestamp when the SMS message was added.

Parameters:

- `$val` - Integer - The new timestamp.

`ReadTimestamp()`

Returns the integer representation of the timestamp when the SMS message was read.

Returns: Integer - The timestamp.

`SetReadTimestamp($val)`

Updates the timestamp when the SMS message was read.

Parameters:

- `$val` - Integer - The new timestamp.

`HasReadTimestamp()`

Checks if the SMS message has a valid read timestamp.

Returns: Boolean - True if the SMS message has a valid read timestamp, false otherwise.

`SendAfterTimestamp()`

Returns the integer representation of the timestamp when the SMS message is scheduled to be sent.

Returns: Integer - The timestamp.

`SetSendAfterTimestamp($val)`

Updates the timestamp when the SMS message is scheduled to be sent.

Parameters:

- `$val` - Integer - The new timestamp.

`HasSendAfterTimestamp()`

Checks if the SMS message has a valid scheduled send timestamp.

Returns: Boolean - True if the SMS message has a valid scheduled send timestamp, false otherwise.

`ReadByClientTimestamp()`

Returns the integer representation of the timestamp when the SMS message was read by the client.

Returns: Integer - The timestamp.

`SetReadByClientTimestamp($val)`

Updates the timestamp when the SMS message was read by the client.

Parameters:

- `$val` - Integer - The new timestamp.

`HasReadByClientTimestamp()`

Checks if the SMS message has a valid read timestamp by the client.

Returns: Boolean - True if the SMS message has a valid read timestamp by the client, false otherwise.

`AssetID()`

Returns the integer representation of the asset ID associated with the SMS message.

Returns: Integer - The asset ID.

`SetAssetID($val)`

Updates the asset ID associated with the SMS message.

Parameters:

- `$val` - Integer - The new asset ID.

`HasAssetID()`

Checks if the SMS message has a valid asset ID.

Returns: Boolean - True if the SMS message has a valid asset ID, false otherwise.

`isUnread()`

Checks if the SMS message is unread.

Returns: Boolean - True if the SMS message is unread, false otherwise.

`SendAsSMSNow($type_organization = false)`

Sends the SMS message as an SMS immediately.

Parameters:

- `$type_organization` - Boolean - Optional. Set to true for organization type.

Create

`CreateOutgoing($text, $company, $customer = 0, $referral = 0, $logtype='', $employee=0, $asset=0, $scheduletime=0, $contact_id=0, $campaign_id=0, $id_campaign_schedule=0, $organization = 0)`

Creates an outgoing SMS message and inserts it into the database.

Type:

- Static method.

Parameters:

- `$text` - String - The content of the SMS message.
- `$company` - Integer - The company ID associated with the SMS message.
- `$customer` - Integer - Optional. The customer ID associated with the SMS message. Default is 0.
- `$referral` - Integer - Optional. The referral ID associated with the SMS message. Default is 0.
- `$logtype` - String - Optional. The type of log for the SMS message. Default is an empty string.
- `$employee` - Integer - Optional. The employee ID associated with the SMS message. Default is 0.
- `$asset` - Integer - Optional. The asset ID associated with the SMS message. Default is 0.
- `$scheduletime` - Integer - Optional. The timestamp when the SMS message is scheduled to be sent. Default is 0.
- `$contact_id` - Integer - Optional. The contact ID associated with the SMS message. Default is 0.
- `$campaign_id` - Integer - Optional. The campaign ID associated with the SMS message. Default is 0.
- `$id_campaign_schedule` - Integer - Optional. The campaign schedule ID associated with the SMS message. Default is 0.
- `$organization` - Integer - Optional. The organization ID associated with the SMS message. Default is 0.

Returns: Integer - The ID of the newly created outgoing SMS message in the database.

MessagesThreadItem class

- Full class name: `\GS\Messages\MessagesThread\MessagesThreadItem`
- Legacy alias (shouldn't be used in a new code): ~~`\TMessagesLog`~~

Information:

- Type: ORM Model
- DB Table: `messagelog`

This class is designed to gather different types of messages as a single thread for the objects within CRM. Its purpose is to represent a single item in a conversation with additional information.

Possible message types:

- email
- sms
- call

Possible objects:

- Customer
- Organization

If neither a customer nor an organization is set, it means that `MessagesThreadItem` represents an internal conversation item.

Table of Contents

General Methods

`ID()`

Returns the ID of the MessagesThreadItem.

Parameters: None.

Returns: Integer.

`Exists()`

Checks if the MessagesThreadItem exists.

Parameters: None.

Returns: Boolean.

`UpdateValues($params)`

Updates specified values for the MessagesThreadItem.

Parameters:

- `$params` - Associative array of values to be updated.

Returns: None.

`Create($type, $id_message, $company, $customer = 0, $time = 0, $incoming='', $id_employee=0, $id_reply = 0, $organization = 0)`

Creates a new MessagesThreadItem. Static method.

Parameters:

- `$type` - Type of the message (string, one of the possible message type constans).
- `$id_message` - Message ID (message ID for a particular object).
- `$company` - Company ID.
- `$customer` - Customer ID (optional, `0` means not attached to any customer).
- `$time` - Timestamp (optional, current timestamp on 0 will be set).
- `$incoming` - Incoming flag (optional).
- `$id_employee` - Employee ID or Employee object (optional, `0` means not attached to any Employee).
- `$id_reply` - ID of the replied message (optional, `0` means it is not a reply to any message).
- `$organization` - Organization ID (optional, `0` means not attached to any organization).

Returns: Created MessagesThreadItem object.

`Remove()`

Removes the MessagesThreadItem from the database.

Parameters: None.

Returns: None.

Initializers

`initWithMessageIDAndType(int $id_message, string $type)`

Initializes a MessagesThreadItem using the message ID and type. Static method.

Parameters:

- `$id_message` - Message ID.
- `$type` - Type of the message.

Returns: MessagesThreadItem object.

`UsingCache($id)`

Creates or retrieves a MessagesThreadItem object using cache if available. Static method.

Parameters:

- `$id` - ID used for creating or retrieving the MessagesThreadItem.

Returns: MessagesThreadItem object.

Message Information

`Type()`

Returns the type of the MessagesThreadItem, as one of the possible message type constans.

Parameters: None.

Returns: String.

`MessageID()`

Returns the message ID associated with the MessagesThreadItem. It is an ID of a particular model:

- SMSMessage
- EmailMessage
- VoiceCall

Parameters: None.

Returns: Integer.

`SetType($val)`

Sets the type of the MessagesThreadItem. Usually you should use both `SetType` and `SetMessageID`.

Parameters:

- `$val` - New type value (one of the possible message type constans).

Returns: None.

`SetMessageID($val)`

Sets the message ID for the MessagesThreadItem. Usually you should use both `SetType` and `SetMessageID`.

Parameters:

- `$val` - New message ID value.

Returns: None.

`isIncoming()`

Checks if the MessagesThreadItem represents an incoming (`true`) or outgoing (`false`) message.

Parameters: None.

Returns: Boolean.

`CompanyID()`

Returns the company ID associated with the MessagesThreadItem.

Parameters: None.

Returns: Integer.

`SetCompanyID(int $val)`

Sets the company ID for the MessagesThreadItem.

Parameters:

- `$val` - New company ID value.

Returns: None.

`EmployeeID()`

Returns the employee ID associated with the MessagesThreadItem.

Parameters: None.

Returns: Integer.

`TimeSent()`

Returns the timestamp (integer unix timestamp) when the message was sent.

Parameters: None.

Returns: Integer.

`TimeRead()`

Returns the timestamp (integer unix timestamp) when the message was read

- by the object (Customer or Organization) if it's outgoing message
- by the Employee if it's incoming message

Parameters: None.

Returns: Integer.

`isReply()`

Return's `true` if the MessagesThreadItem is a reply.

Parameters: None.

Returns: Boolean.

`SetTimeRead(int $time)`

Sets the timestamp (integer unix timestamp) when the message was read.

Parameters:

- `$time` - New timestamp value.

Returns: None.

`isUnread()`

Return's `true` if the MessagesThreadItem is unread.

Parameters: None.

Returns: Boolean.

Object Information

`CustomerID()`

Returns the customer ID associated with the MessagesThreadItem.

Parameters: None.

Returns: Integer.

`OrganizationID()`

Returns the organization ID associated with the MessagesThreadItem.

Parameters: None.

Returns: Integer.

`SetCustomerID($val)`

Sets the customer ID for the MessagesThreadItem.

Parameters:

- `$val` - New customer ID value.

Returns: None.

MessagesThreadList class

- Full class name: `\GS\Messages\MessagesThread\MessagesThreadList`
- Legacy alias (shouldn't be used in a new code): ~~`\TMessagesLogList`~~

Information:

- Type: ORM Model List
- DB Table: `messagelog`
- List Items: \GS\Messages\MessagesThread\MessagesThreadItem.md

This class is designed to retrieve MessagesThreadItem objects from the database (usually for conversations).

Table of Contents

Type Filters

`SetFilterEmail()`

Sets a filter for the conversation thread to include only email messages.

`SetFilterSMS()`

Sets a filter for the conversation thread to include only SMS messages.

`SetFilterCalls()`

Sets a filter for the conversation thread to include only call-type messages.

`SetFilterExcludeCalls()`

Sets a filter for the conversation thread to exclude call-type messages.

Deprecated alias: `SetFilterMessages`

Object Filters

`SetFilterCustomer($customer_or_id)`

Sets a filter for the conversation thread based on the specified customer identifier.

Parameters:

- `$customer_or_id` - Customer identifier or object.

`SetFilterOrganization($organization_or_id)`

Sets a filter for the conversation thread based on the specified organization identifier.

Parameters:

- `$organization_or_id` - Organization identifier.

Other Filters

`SetFilterCompany($company_or_id)`

Sets a filter for the conversation thread based on the specified company identifier.

Parameters:

- `$company_or_id` - Company identifier or object.

`SetFilterTimeSent($time)`

Sets a filter for the conversation thread to include only messages sent at a specific time.

Parameters:

- `$time` - Time as integer unix timestamp.

`SetFilterNewerThenTimeSent($time)`

Sets a filter for the conversation thread to include only messages sent after a specific time.

Parameters:

- `$time` - Time as integer unix timestamp.

`SetFilterOutgoing()`

Sets a filter for the conversation thread to include only outgoing messages.

`SetFilterIncoming()`

Sets a filter for the conversation thread to include only incoming messages.

`SetFilterGreaterThenID($last_id)`

Sets a filter for the conversation thread to include only messages with an ID greater than the specified value.

Parameters:

- `$last_id` - Message ID.

`SetFilterNotReply()`

Sets a filter for the conversation thread to include only messages that are not replies.

`SetFilterReply($reply_id)`

Sets a filter for the conversation thread to include only messages that are replies to a specific message.

Parameters:

- `$reply_id` - ID of the message being replied to.

`SetFilterUnread()`

Sets a filter for the conversation thread to include only unread messages.

Parameters:

SetFilterWeek()

Sets a filter for the conversation thread to include only messages sent within the current week.

Parameters:

SetFilterAddedTimeBetween($from, $to)

Sets a filter for the conversation thread to include only messages sent within a specific time range.

Parameters:

- `$from` - Start of the time range.
- `$to` - End of the time range.

SetFilterDay()

Sets a filter for the conversation thread to include only messages sent within the current day.

Parameters:

SetFilterMonth()

Sets a filter for the conversation thread to include only messages sent within the current month.

Parameters:

SetFilter7days()

Sets a filter for the conversation thread to include only messages sent within the last 7 days.

Parameters:

SetFilter14days()

Sets a filter for the conversation thread to include only messages sent within the last 14 days.

Parameters:

SetFilter30days()

Sets a filter for the conversation thread to include only messages sent within the last 30 days.

Parameters:

SetFilterCustomersIDs(array $arrayids)

Sets a filter for the conversation thread to include only messages related to specified customer IDs.

Parameters:

- `$arrayids` - Array of customer IDs.

Returns:

Ordering

`OrderByID($asc=true)`

Orders the conversation thread by ID, in ascending or descending order.

Parameters:

- `$asc` - (Optional) If true, orders in ascending order. Default is true.

`OrderByTimestamp($asc=true)`

Orders the conversation thread by timestamp, in ascending or descending order.

Parameters:

- `$asc` - (Optional) If true, orders in ascending order. Default is true.

Organization class

- Full class name: `\GS\Organization\Organization`
- Legacy alias (shouldn't be used in a new code): ~~`\TOrganization`~~

Information:

- Type: ORM Model
- DB Table: `organizations`

This class represents every organization for a business (company, NGO, government agency etc.).

Table of Contents

Common

`ID()`

Gets the ID of the organization.

Returns: The ID of the organization.

`Exists()`

Checks if the organization exists.

Returns: Boolean indicating whether the organization exists.

`Title()`

Returns the title of the organization.

`UpdateValues($params)`

Updates the values of the organization.

Parameters:

- `$params` - Array, key-value pairs of fields and their new values

Returns: None

`Remove()`

Removes the organization and associates customers.

Parameters: None

Returns: None

`SetTitle($val)`

Sets the title of the organization.

Parameters:

- `$val` - The new title.

`HasTitle()`

Checks if the organization has a title.

Returns: Boolean

`DetailsURL()`

Returns the URL for the organization details page.

`SetLastUpdateTime()`

Sets the last update time for the organization.

`Initials()`

Returns the initials of the organization based on the title.

`Create()`

Creates a new organization.

Parameters: None

Returns: Object (new organization)

`Disable()`

Disables the organization.

Parameters: None

Returns: `void`

`Enable()`

Enables the organization.

Parameters: None

Returns: `void`

`isEnabled()`

Checks if the organization is enabled.

Parameters: None

Returns: `bool`

Metadata

`SetMetaData($key, $val)`

Sets metadata for the organization.

Parameters:

- `$key` - Key for the metadata.
- `$val` - Value of the metadata.

Returns: `void`

`GetMetaData($key)`

Gets metadata for the organization.

Parameters:

- `$key` - Key for the metadata.

Returns: `mixed` (Value of the metadata or null if not found)

Initializers

`initWithRawPhonesArray($phones, $id_company)`

Initializes an organization using an array of raw phone numbers.

Parameters:

- `$phones` - An array of phone numbers.
- `$id_company` - The ID of the company.

Returns: An instance of the `OrganizationCellPhone` class if the organization exists; otherwise, an empty string.

`initWithEmailsArray($emails, $id_company)`

Initializes an organization using an array of email addresses.

Parameters:

- `$emails` - An array of email addresses.
- `$id_company` - The ID of the company.

Returns: An instance of the `Organization` class if the organization exists; otherwise, an empty string.

`initWithPhone($phone, $id_company, $country = '')`

Initializes an organization using a single phone number.

Parameters:

- `$phone` - The phone number.
- `$id_company` - The ID of the company.
- `$country` - (Optional) The country of the phone number.

Returns: An instance of the `OrganizationCellPhone` class.

`initWithPhonesArray($phones, $id_company)`

Initializes an organization using an array of phone numbers with country information.

Parameters:

- `$phones` - An array of phone numbers with associated country information.
- `$id_company` - The ID of the company.

Returns: An instance of the `OrganizationCellPhone` class if the organization exists; otherwise, an empty string.

`initWithTitle($title, $id_company)`

Initializes an organization using its title and the ID of the company.

Parameters:

- `$title` - The title of the organization.
- `$id_company` - The ID of the company.

Returns: An instance of the `Organization` class.

`getByEmail($email)`

Retrieves an organization using its email address.

Parameters:

- `$email` - The email address of the organization.

Returns: An instance of the `TCompany` class if the organization exists; otherwise, `false`.

`initWithEmail($email, $id_company)`

Initializes an organization using its email address and the ID of the company.

Parameters:

- `$email` - The email address of the organization.
- `$id_company` - The ID of the company.

Returns: An instance of the `Organization` class.

`UsingCache($id)`

Retrieves an organization from the cache or creates a new instance if not cached.

Parameters:

- `$id` - The ID of the organization.

Returns: An instance of the `Organization` class.

Tags

`HasTagID($tag_id)`

Checks if the organization has a specific tag.

Parameters:

- `$tag_id` - The ID of the tag.

Returns: Boolean indicating whether the organization has the specified tag.

`TagsCount()`

Gets the count of tags associated with the organization.

Returns: The count of tags.

`GetTagsIds()`

Gets an array of tag IDs associated with the organization.

Returns: An array of tag IDs.

`SetTagID($tag_id)`

Sets a tag for the organization.

Parameters:

- `$tag_id` - The ID of the tag.

`UnsetTagID($tag_id)`

Removes a tag from the organization.

Parameters:

- `$tag_id` - The ID of the tag.

`StageTag()`

Returns the stage tag of the organization.

`HasStageTag()`

Checks if the organization has a stage tag.

Returns: Boolean

`SetStageTag($tagname)`

Sets the stage tag for the organization.

Parameters:

- `$tagname` - The new stage tag.

`LoadTags($rewritecache = false)`

Loads tags for the organization.

Parameters:

- `$rewritecache` - (Optional) Boolean flag to force rewriting cache.

Company

`CompanyID()`

Gets the ID of the company associated with the organization.

Returns: The ID of the company.

`SetCompanyID($val)`

Sets the ID of the company for the organization.

Parameters:

- `$val` - The ID of the company.

`HasCompanyID()`

Checks if the organization has a company ID.

Returns: Boolean indicating whether the organization has a company ID.

Stage

`StageInfoDecoration($status_tag)`

Returns HTML for displaying the stage information decoration.

Parameters:

- `$status_tag` - The status tag for filtering stages.

Returns: HTML

`StageInfo($status_tag)`

Returns an array with total count and active stage for a given status tag.

Parameters:

- `$status_tag` - The status tag for filtering stages.

Returns: Array

Pipeline

`PipelineTypeTitle()`

Returns the title based on the pipeline type.

`HasPipelineType()`

Checks if the organization has a pipeline type.

Returns: Boolean

`PipelineType()`

Returns the pipeline type of the organization.

`SetPipelineType($value)`

Sets the pipeline type for the organization.

Parameters:

- `$value` - The new pipeline type.

Sort Order

`SortOrder()`

Returns the sort order of the organization.

`SetSortOrder($val)`

Sets the sort order for the organization.

Parameters:

- `$val` - The new sort order.

Address and Contacts

`CellphoneFormatted()`

Returns the formatted cellphone number.

`Cellphone()`

Returns the cellphone number.

`MainCellphoneID()`

Returns the ID of the main cellphone.

`isMainCellphoneActive()`

Checks if the main cellphone is active.

Returns: Boolean

`CellphoneOLD()`

Returns the old cellphone number.

`SetCellphone($val)`

Sets the cellphone number for the organization.

Parameters:

- `$val` - The new cellphone number.

`HasCellphone()`

Checks if the organization has a cellphone.

Returns: Boolean

`HasCellphoneOLD()`

Checks if the organization has an old cellphone number.

Returns: Boolean

`CreateMainCellPhone($val)`

Creates the main cellphone for the organization.

Parameters:

- `$val` - The new cellphone number.

`Email()`

Returns the email of the organization.

`SetEmail($val)`

Sets the email for the organization.

Parameters:

- `$val` - The new email.

`HasEmail()`

Checks if the organization has an email.

Returns: Boolean

`HasFullAddressAndName()`

Checks if the organization has a title, address, city, and zip code.

Returns: Boolean indicating whether the organization has full address and name details.

`AddressFormatted1()`

Returns a formatted version of address including city, state, and zip.

`AddressFormatted()`

Returns a formatted version of the address.

`Address()`

Returns the address of the organization.

`SetAddress($val)`

Sets the address for the organization.

Parameters:

- `$val` - The new address.

`HasAddress()`

Checks if the organization has an address.

Returns: Boolean

`City()`

Returns the city of the organization.

`SetCity($val)`

Sets the city for the organization.

Parameters:

- `$val` - The new city.

`HasCity()`

Checks if the organization has a city.

Returns: Boolean

`State()`

Returns the state of the organization.

`SetState($val)`

Sets the state for the organization.

Parameters:

- `$val` - The new state.

`HasState()`

Checks if the organization has a state.

Returns: Boolean

`Zip()`

Returns the zip code of the organization.

`SetZip($val)`

Sets the zip code for the organization.

Parameters:

- `$val` - The new zip code.

`HasZip()`

Checks if the organization has a zip code.

Returns: Boolean

`Country()`

Returns the country of the organization.

`SetCountry($val)`

Sets the country for the organization.

Parameters:

- `$val` - Country value.

Returns: N/A

`HasCountry()`

Checks if the organization has a country.

Parameters: N/A

Returns:

- Boolean indicating whether the country information is present.

Business Category

`BusinessCategory()`

Gets the business category of the organization.

Parameters: N/A

Returns:

- Business category value.

`SetBusinessCategory($val)`

Sets the business category for the organization.

Parameters:

- `$val` - Business category value.

Returns: N/A

`HasBusinessCategory()`

Checks if the organization has a business category.

Parameters: N/A

Returns:

- Boolean indicating whether the business category information is present.

SIC-Code

`SicCode()`

Gets the SIC code of the organization.

Parameters: N/A

Returns:

- SIC code value.

`SetSicCode($val)`

Sets the SIC code for the organization.

Parameters:

- `$val` - SIC code value.

Returns: N/A

`HasSicCode()`

Checks if the organization has an SIC code.

Parameters: N/A

Returns:

- Boolean indicating whether the SIC code information is present.

URLs

`Url()`

Gets the URL of the organization.

Parameters: N/A

Returns:

- URL value.

`SetUrl($val)`

Sets the URL for the organization.

Parameters:

- `$val` - URL value.

Returns: N/A

`HasUrl()`

Checks if the organization has a URL.

Parameters: N/A

Returns:

- Boolean indicating whether the URL information is present.

`Facebook()`

Gets the Facebook profile of the organization.

Parameters: N/A

Returns:

- Facebook profile value.

`SetFacebook($val)`

Sets the Facebook profile for the organization.

Parameters:

- `$val` - Facebook profile value.

Returns: N/A

`HasFacebook()`

Checks if the organization has a Facebook profile.

Parameters: N/A

Returns:

- Boolean indicating whether the Facebook profile information is present.

`Twitter()`

Gets the Twitter handle of the organization.

Parameters: N/A

Returns:

- Twitter handle value.

`SetTwitter($val)`

Sets the Twitter handle for the organization.

Parameters:

- `$val` - Twitter handle value.

Returns: N/A

`HasTwitter()`

Checks if the organization has a Twitter handle.

Parameters: N/A

Returns:

- Boolean indicating whether the Twitter handle information is present.

`LinkedIn()`

Gets the LinkedIn profile of the organization.

Parameters: N/A

Returns:

- LinkedIn profile value.

`SetLinkedIn($val)`

Sets the LinkedIn profile for the organization.

Parameters:

- `$val` - LinkedIn profile value.

Returns: N/A

`HasLinkedIn()`

Checks if the organization has a LinkedIn profile.

Parameters: N/A

Returns:

- Boolean indicating whether the LinkedIn profile information is present.

`Instagram()`

Gets the Instagram profile of the organization.

Parameters: N/A

Returns:

- Instagram profile value.

`SetInstagram($val)`

Sets the Instagram profile for the organization.

Parameters:

- `$val` - Instagram profile value.

Returns: N/A

`HasInstagram()`

Checks if the organization has an Instagram profile.

Parameters: N/A

Returns:

- Boolean indicating whether the Instagram profile information is present.

`HasSocialURLS()`

Checks if the organization has any social media profiles (Facebook, Twitter, Instagram, LinkedIn).

Parameters: N/A

Returns:

- Boolean indicating whether any of the social media profiles are present.

Value

`Value()`

Gets the value of the organization.

Parameters: N/A

Returns:

- Float value indicating the organization's value.

`HasValue()`

Checks if the organization has a value.

Parameters: N/A

Returns:

- Boolean indicating whether the value information is present.

`SetValue($value)`

Sets the value for the organization.

Parameters:

- `$value` - Float value representing the organization's value.

Returns: N/A

Indicators

`GetARR()`

Gets the Annual Recurring Revenue (ARR) of the organization.

Parameters: N/A

Returns:

- Formatted string representing the ARR.

`GetMRR()`

Gets the Monthly Recurring Revenue (MRR) of the organization.

Parameters: N/A

Returns:

- Formatted string representing the MRR.

`EmployeeSizeFrom()`

Gets the starting range of the employee size for the organization.

Parameters: N/A

Returns: Integer value indicating the starting range of the employee size.

`HasEmployeeSizeFrom()`

Checks if the organization has a starting range for employee size.

Parameters: N/A

Returns:

- Boolean indicating whether the starting range for employee size information is present.

`SetEmployeeSizeFrom($value)`

Sets the starting range for employee size for the organization.

Parameters:

- `$value` - Integer value representing the starting range for employee size.

Returns: N/A

`HasEmployeeSize()`

Checks if the organization has employee size information.

Parameters: N/A

Returns:

- Boolean indicating whether the employee size information is present.

`EmployeeSize()`

Calculates and returns the employee size range for the organization.

Parameters: N/A

Returns:

- String representing the employee size range.

`EmployeeSizeTo()`

Gets the ending range of the employee size for the organization.

Parameters: N/A

Returns:

- Integer value indicating the ending range of the employee size.

`HasEmployeeSizeTo()`

Checks if the organization has an ending range for employee size.

Parameters: N/A

Returns:

- Boolean indicating whether the ending range for employee size information is present.

`SetEmployeeSizeTo($value)`

Sets the ending range for employee size for the organization.

Parameters:

- `$value` - Integer value representing the ending range for employee size.

Returns: N/A

Registration Information

`AddedByEmployeeID()`

Alias for `Addedby()`. Returns the ID of the employee who added the organization.

Parameters: N/A

Returns:

- Integer value representing the ID of the employee who added the organization.

`Addedby()`

Gets the ID of the employee who added the organization.

Parameters: N/A

Returns:

- Integer value representing the ID of the employee who added the organization.

`SetAddedby($val, $assign_contacts = false)`

Sets the ID of the employee who added the organization. Optionally, assigns the same employee ID to related contacts.

Parameters:

- `$val` - Integer value representing the employee ID.
- `$assign_contacts` - Optional. Boolean indicating whether to assign the employee ID to related contacts.

Returns: N/A

`HasAddedby()`

Checks if the organization has an added-by employee ID.

Parameters: N/A

Returns:

- Boolean indicating whether the added-by employee ID information is present.

`AddedTime()`

Returns the added time of the organization.

Parameters: None

Returns: Integer

`SetAddedTime($val)`

Sets the added time for the organization.

Parameters:

- `$val` - Integer, the new added time value

Returns: None

`HasAddedTime()`

Checks if the organization has an added time.

Parameters: None

Returns: Boolean

Phone Validation

`isPhoneValidationForApprove()`

Checks if phone validation is required for approval.

Parameters: None

Returns: Boolean

`SetPhoneValidationForApprove($val)`

Sets the phone validation status for approval.

Parameters:

- `$val` - Boolean, true to enable phone validation, false to disable

Returns: None

`UnsetPhoneValidationForApprove()`

Disables phone validation for approval.

Parameters: None

Returns: None

`HasPhoneValidationForApprove()`

Checks if phone validation for approval is set.

Parameters: None

Returns: Boolean

Opportunity

`GetOpportunityOwner()`

Gets the opportunity owner associated with the organization.

Parameters: None

Returns: Object (opportunity owner organization) or empty string

`OpportunityStageOwner()`

Returns the ID of the opportunity stage owner.

Parameters: None

Returns: Integer

`HasOpportunityStageOwner()`

Checks if there is an opportunity stage owner.

Parameters: None

Returns: Boolean

`SetOpportunityStageOwner($value)`

Sets the opportunity stage owner.

Parameters:

- `$value` - Integer, ID of the new opportunity stage owner

Returns: None

Close Deal

`DealCloseDate()`

Returns the deal close date.

Parameters: None

Returns: Integer

`HasDealCloseDate()`

Checks if there is a deal close date.

Parameters: None

Returns: Boolean

`SetDealCloseDate($value)`

Sets the deal close date.

Parameters:

- `$value` - Integer, new deal close date

Returns: None

`SetConfidenceLevel($value)`

Sets the confidence level for the organization.

Parameters:

- `$value` - Integer, new confidence level

Returns: None

Confidence Level

`HasConfidenceLevel()`

Checks if there is a confidence level set.

Parameters: None

Returns: Boolean

`GetConfidenceLevel()`

Gets the confidence level of the organization.

Parameters: None

Returns: Integer

Customers

`GetCustomersCount()`

Gets the count of customers associated with the organization.

Parameters: None

Returns: Integer

`GetCustomersIDs()`

Gets the IDs of customers associated with the organization.

Parameters: None

Returns: Array of integers (customer IDs)

Taxonomy

`GetTaxonomy($object_name, $object_id, $use_object_id_as_rel_id=false)`

Gets taxonomy associated with the organization.

Parameters:

- `$object_name` - String, name of the taxonomy object
- `$object_id` - Integer, ID of the taxonomy object
- `$use_object_id_as_rel_id` - Boolean, whether to use object ID as relationship ID

Returns: Array of integers (taxonomy IDs)

`SetTaxonomy($object_name, $object_id, $rel_id)`

Sets taxonomy for the organization.

Parameters:

- `$object_name` - String, name of the taxonomy object
- `$object_id` - Integer, ID of the taxonomy object
- `$rel_id` - Integer, relationship ID for taxonomy

Returns: None

`UnsetTaxonomy($object_name, $object_id, $rel_id)`

Unsets taxonomy for the organization.

Parameters:

- `$object_name` - String, name of the taxonomy object
- `$object_id` - Integer, ID of the taxonomy object
- `$rel_id` - Integer, relationship ID for taxonomy

Returns: None

Last Messages

`GetLastEmail()`

Gets the last email log associated with the organization.

Parameters: None

Returns: `TCustomerLog` or `null`

`GetLastSMS()`

Gets the last SMS log associated with the organization.

Parameters: None

Returns: `TCustomerLog` or `null`

`GetLastNote()`

Gets the last note associated with the organization.

Parameters: None

Returns: `TCustomerLog` or `false` if none

`GetLastCall()`

Gets the last call associated with the organization.

Parameters: None

Returns: `TCustomerLog` or `null`

Searches

`SetSearchID($search_id)`

Sets the search ID for the organization.

Parameters:

- `$search_id` - Search ID to set.

Returns: `void`

`UnsetSearchID($search_id)`

Unsets the specified search ID for the organization.

Parameters:

- `$search_id` - Search ID to unset.

Returns: `void`

`UnsetSearchIDs()`

Unsets all search IDs associated with the organization.

Parameters: None

Returns: `void`

`SearchCount()`

Gets the count of search IDs associated with the organization.

Parameters: None

Returns: `int`

`LoadSearches($rewritecache = false)`

Loads search IDs for the organization.

Parameters:

- `$rewritecache` - Whether to rewrite the cache.

Returns: `void`

Campaigns

`LoadCampaigns($rewritecache = false)`

Loads campaigns associated with the organization.

Parameters:

- `$rewritecache` - Whether to rewrite the cache.

Returns: `void`

`UnsetCampaigns()`

Unsets all campaigns associated with the organization.

Parameters: None

Returns: `void`

`AddToCampaign($campaign)`

Adds the organization to a specified campaign.

Parameters:

- `$campaign` - Campaign to add the organization to.

Returns: `void`

`RemoveFromCampaign($campaign)`

Removes the organization from a specified campaign.

Parameters:

- `$campaign` - Campaign to remove the organization from.

Returns: `void`

Attachments

`LoadAttachments($rewritecache)`

Loads attachments associated with the organization.

Parameters:

- `$rewritecache` - Whether to rewrite the cache and force reloading attachments.

Returns: `void`

`HasAttachments()`

Checks if the organization has any attachments.

Parameters: None

Returns: `bool` - Returns true if the organization has attachments; otherwise, false.

`GetAttachmentsList()`

Gets a list of attachments associated with the organization.

Parameters: None

Returns: `array` - An array of attachment information, each containing:

- `id` - Attachment ID
- `title` - Uploaded file name
- `url` - URL of the attachment thumbnail
- `download_url` - URL for downloading the attachment

`UnsetAttachments()`

Unsets (removes) the attachments associated with the organization.

Parameters: None

Returns: `void`

`ClearAttachments()`

Clears (removes) all attachments associated with the organization.

Parameters: None

Returns: `void`

`UnsetAttachmentID($id)`

Unsets (removes) a specific attachment by ID.

Parameters:

- `$id` - ID of the attachment to be removed.

Returns: `void`

`AddAttachment($attachment)`

Adds an attachment to the organization.

Parameters:

- `$attachment` - Attachment ID to be added.

Returns: `void`

`MarkAsConversationUnread()`

Marks the organization conversation as unread.

Parameters: None

Returns: `void`

Conversations

`MarkAsConversationRead()`

Marks the conversation as read.

Parameters: None

Returns: None

`HasUnreadConversation()`

Checks if the conversation has unread messages.

Parameters: None

Returns: `boolean`

Unsubscribe

`SetUnsubscribeFromTextStatus($val)`

Sets the unsubscribe status for text messages.

Parameters:

- `$val` - Unsubscribe status value.

Returns: None

`GetUnsubscribeFromTextStatus()`

Gets the unsubscribe status for text messages.

Parameters: None

Returns: `string`

`isUnsubscribeFromTextStatus()`

Checks if the organization is unsubscribed from text messages.

Parameters: None

Returns: `boolean`

`GetUnsubscribeStatus()`

Gets the general unsubscribe status.

Parameters: None

Returns: `string`

`isUnsubscribeStatus()`

Checks if the organization is unsubscribed.

Parameters: None

Returns: `boolean`

Accept Information

`GetSMSAcceptedTimestamp()`

Gets the timestamp when SMS was accepted.

Parameters: None

Returns: `integer`

`SetSMSAcceptedTimestamp($val=NULL)`

Sets the timestamp when SMS was accepted.

Parameters:

- `$val` - Timestamp value (optional).

Returns: None

`GetSMSAcceptedTag()`

Gets the SMS accepted tag.

Parameters: None

Returns: `string`

`SetSMSAcceptedTag($val)`

Sets the SMS accepted tag.

Parameters:

- `$val` - SMS accepted tag value.

Returns: None

`GetSMSPendingTimestamp()`

Gets the timestamp for pending SMS.

Parameters: None

Returns: `integer`

`SetSMSPendingTimestamp($val=NULL)`

Sets the timestamp for pending SMS.

Parameters:

- `$val` - Timestamp value (optional).

Returns: None

Sending Messages

`isSendingMessagesUndefined()`

Checks if sending messages status is undefined.

Parameters: None

Returns: `boolean`

`isSendingMessagesRejected()`

Checks if sending messages status is rejected.

Parameters: None

Returns: `boolean`

`isSendingMessagesNoResponse()`

Checks if there is no response for sending messages.

Parameters: None

Returns: `boolean`

`isSendingMessagesPending()`

Checks if sending messages are pending.

Parameters: None

Returns: `boolean`

`SendingMessagesError()`

Gets the error message related to sending messages.

Parameters: None

Returns: `string`

`isSendingMessagesEnabled()`

Checks if sending messages are enabled.

Parameters: None

Returns: `boolean`

`isSendingMessagesAbilityExpired()`

Checks if the ability to send messages is expired.

Parameters: None

Returns: `boolean`

`SendMessage($text, $id_employee = 0, $writelog = true, $logtype = 'conversation', $scheduletime = 0, $attachments = Array(), $send_message_as_sms = true, $asset_id = 0, $contact_id = 0, $campaign_id = 0, $id_campaign_schedule = 0)`

Sends a message.

Parameters:

- `$text` - Message text.
- `$id_employee` - Employee ID.
- `$writelog` - Boolean flag for writing to the log.
- `$logtype` - Log type.
- `$scheduletime` - Schedule time for sending.
- `$attachments` - Array of attachments.
- `$send_message_as_sms` - Boolean flag for sending as SMS.
- `$asset_id` - Asset ID.
- `$contact_id` - Contact ID.
- `$campaign_id` - Campaign ID.
- `$id_campaign_schedule` - Campaign schedule ID.

Returns: `integer` (Message ID)

---

protected function SendSMS($text, $messagetype = 'conversation', $scheduletime = 0, $attachments = Array(), $contact_id = 0, $campaign_id = 0, $id_employee = 0)

Sends an SMS.

Parameters:

- `$text` - SMS text.
- `$messagetype` - Message type.
- `$scheduletime` - Schedule time for sending.
- `$attachments` - Array of attachments.
- `$contact_id` - Contact ID.
- `$campaign_id` - Campaign ID.
- `$id_employee` - Employee ID.

Returns: None

SendEmailFromEmployee($subject, $message, $scheduletime = 0, $writelog = false, $employee = 0, $campaign = 0, $campaign_item = 0, $email_id = '', $cc = '', $bcc = '')

Sends an email from an employee.

Parameters:

- `$subject` - Email subject.
- `$message` - Email message.
- `$scheduletime` - Schedule time for sending.
- `$writelog` - Boolean flag for writing to the log.
- `$employee` - Employee object or ID.
- `$campaign` - Campaign ID.
- `$campaign_item` - Campaign item ID.
- `$email_id` - Email ID for reply.
- `$cc` - CC email address.
- `$bcc` - BCC email address.

Returns: `integer` (Email ID) or `boolean`

SendEmailFromCompany($subject, $message, $scheduletime = 0, $writelog = false, $employee = 0, $campaign = 0, $campaign_item = 0, $bulk = false, $id_campaign_schedule = 0, $email_id = '', $cc = '', $bcc = '')

Sends an email from the company on behalf of the organization.

Parameters:

- `$subject` - Subject of the email.
- `$message` - Body of the email.
- `$scheduletime` - Scheduled time for sending the email.
- `$writelog` - Whether to write a log entry.
- `$employee` - Employee associated with the email.
- `$campaign` - Campaign ID associated with the email.
- `$campaign_item` - Campaign item associated with the email.
- `$bulk` - Whether the email is sent in bulk.
- `$id_campaign_schedule` - Campaign schedule ID.
- `$email_id` - Email ID for reply.
- `$cc` - CC recipients.
- `$bcc` - BCC recipients.

Returns: `bool` (true if successful)

SendInitialSMS()

Sends an initial SMS to the organization for compliance purposes.

Parameters:

Returns:

ReSendInitialSMS()

Resends the initial SMS to the organization.

Parameters: None

Returns: None

Miscellaneous

IncomingSMS(text, id_employee)

Processes an incoming SMS, logs it, and triggers notifications.

Parameters:

- `text` - The incoming SMS text.

- `id_employee` - The ID of the employee handling the SMS.

Returns: None

`MergeAndRemove($organization_id)`

Merges with another organization and removes the current one.

Parameters:

- `$organization_id` - Integer, ID of the organization to merge with

Returns: None

`UpdateByProxyCurlDetails($arr = [])`

Updates organization details using proxy cURL details.

Parameters:

- `$arr` - Array containing details for update.

Returns: `void`

`UnreadEmailsCount()`

Gets the count of unread emails.

Parameters: None

Returns: `integer`

Organization List class

- Full class name: `\GS\Organization\OrganizationList`
- Legacy alias (shouldn't be used in a new code): ~~`\TOrganizationList`~~

Information:

- Type: ORM Model List
- DB Table: `organizations`
- List Items: \GS\Organization\Organization

Table of Contents:

- Aggregation

Filtering

`SetFilterCompany($company_or_id)`

Sets a filter based on the organization's company ID.

Parameters:

- `$company_or_id` - Organization's company ID.

`SetFilterTitleBeginnig(string $beginning)`

Sets a filter to include organizations with titles starting with the specified string.

Parameters:

- `$beginning` - The beginning string of the organization title.

`SetFilterTitleOrEmail(string $beginning)`

Sets a filter to include organizations with titles or emails containing the specified string.

Parameters:

- `$beginning` - The string to be included in the organization title or email.

`SetFilterEmailBeginnig(string $beginning)`

Sets a filter to include organizations with email addresses starting with the specified string.

Parameters:

- `$beginning` - The beginning string of the organization email.

`SetFilterCityBeginnig(string $beginning)`

Sets a filter to include organizations with cities starting with the specified string.

Parameters:

- `$beginning` - The beginning string of the organization city.

`SetFilterStateBeginnig(string $beginning)`

Sets a filter to include organizations with states starting with the specified string.

Parameters:

- `$beginning` - The beginning string of the organization state.

`SetFilterValue(int $min, int $max)`

Sets a filter to include organizations with values falling within the specified range.

Parameters:

- `$min` - Minimum value.
- `$max` - Maximum value.

`SetFilterState(string $state)`

Sets a filter to include organizations with a specific state.

Parameters:

- `$state` - The state to filter by.

`SetFilterEnabled()`

Sets a filter to include only enabled organizations.

`SetFilterEmployeeSizeTo(int $size)`

Sets a filter to include organizations with an employee size up to the specified value.

Parameters:

- `$size` - Maximum employee size.

`SetFilterCustomersPipeline()`

Sets a filter to include organizations in the **Customer** pipeline.

`SetFilterPipelineTag($status_tag)`

Sets a filter to include organizations with a specific pipeline tag.

Parameters:

- `$status_tag` - The pipeline tag to filter by.

`SetFilterOpportunitiesPipeline()`

Sets a filter to include organizations in the **Opportunity** pipeline.

`SetFilterLeadsPipeline()`

Sets a filter to include organizations in the **Lead** pipeline.

`SetFilterAddedBy($employee_or_id)`

Sets a filter to include organizations added by a specific employee.

Parameters:

- `$employee_or_id` - Employee ID.

`SetFilterStage($stage)`

Sets a filter to include organizations with a specific stage.

Parameters:

- `$stage` - The stage to filter by.

`SetFilterDay()`

Sets a filter to include organizations added on the current day.

`SetFilterWeek()`

Sets a filter to include organizations added in the current week.

`SetFilter7days()`

Sets a filter to include organizations added in the last 7 days.

`SetFilter14days()`

Sets a filter to include organizations added in the last 14 days.

`SetFilter30days()`

Sets a filter to include organizations added in the last 30 days.

`SetFilterMonth()`

Sets a filter to include organizations added in the current month.

`SetFilter10DaysOld()`

Sets a filter to include organizations updated within the last 10 days.

`SetFilterConfidenceLevel($value)`

Sets a filter to include organizations with a specific confidence level.

Parameters:

- `$value` - The confidence level to filter by.

`SetFilterCellPhoneLike(string $like_condition)`

Sets a filter to include organizations with cell phone numbers matching the specified condition.

Parameters:

- `$like_condition` - The condition to match against cell phone numbers.

`SetFilterCityLike(string $city)`

Sets a filter to include organizations with cell phone numbers matching the specified city.

Parameters:

- `$city` - The city to match against cell phone numbers.

`SetFilterEmailBeginning(string $beginning)`

Sets a filter to include organizations with email addresses starting with the specified string.

Parameters:

- `$beginning` - The beginning string of the organization email.

`SetFilterSMSAcceptPendingTimeLTE(?int $time=NULL)`

Sets a filter to include organizations with SMS pending time less than or equal to the specified time.

Parameters:

- `$time` - (Optional) The time threshold. Defaults to the current time. Time is integer unix timestamp.

`SetFilterSMSAcceptedTag($tag)`

Sets a filter to include organizations with a specific SMS acceptance tag.

Parameters:

- `$tag` - The SMS acceptance tag to filter by.

`SetFilterSMSAcceptPending1()`

Sets a filter to include organizations with SMS acceptance tag 'pending1'.

`SetFilterSMSAcceptPending2()`

Sets a filter to include organizations with SMS acceptance tag 'pending2'.

`SetFilterSMSAcceptNoResponse()`

Sets a filter to include organizations with SMS acceptance tag 'noresponse'.

Ordering

`OrderBySortOrder()`

Orders the organizations by sort order specificied manually in the CRM interface.

`OrderByValues($asc=true)`

Orders the organizations by their values.

Parameters:

- `$asc` - (Optional) Set to false for descending order.

`OrderByConfidenceLevel($asc=true)`

Orders the organizations by confidence level.

Parameters:

- `$asc` - (Optional) Set to false for descending order.

`OrderByDealCloseDate($asc=true)`

Orders the organizations by deal close date.

Parameters:

- `$asc` - (Optional) Set to false for descending order.

`OrderByMultipleField($orderByStr)`

Orders the organizations based on a custom order specified by the `$orderByStr` parameter. MySQL ordering fromat required.

Parameters:

- `$orderByStr` - Custom order string.

Aggregation

`TotalValue($rewritecache=false)`

Returns the total value of organizations based on the applied filters.

Parameters:

- `$rewritecache` - (Optional) Set to true to force cache rewriting.

Returns:

- Total value of organizations.

PendingSMSMessage class

- Full class name: `\GS\Messages\SMS\PendingSMSMessage`
- Legacy alias (shouldn't be used in new code): ~~`\TPendingMessage`~~

Information:

- Type: ORM Model
- DB Table: `smslog_pending`
- List: \GS\Messages\SMS\PendingSMSMessageList

This class is designed to work with pending SMS messages. All SMS that were sent to some phone numbers should be in *pending* status unless recipients confirm that they want to receive the messages. Pending messages won't appear in regular messages conversations.

This class is descendant of \GS\Messages\SMS\GenericSMS, and it has all parent's methods.

PendingSMSMessageList class

- `\GS\Messages\SMS\PendingSMSMessageList`
- Legacy alias (shouldn't be used in a new code): ~~`\TPendingMessageList`~~

Information:

- Type: ORM Model List
- DB Table: `smslog_pending`

- List Items: \GS\Messages\SMS\PendingSMSMessage

This class is designed to work with list of pending SMS.

This class has all the same methods as it's parent \GS\Messages\SMS\SMSMessageList

SmartSearch class

- Full class name: `\GS\SmartSearch\SmartSearch`

Information:

- Type: ORM Model
- DB Table: `smartsearches`

This class represents the single search template, saved to the database.

Table of Contents

Common

`__construct(id_or_cachedinfo)`

Constructs a new SmartSearch object. Accepts either an ID or cached information array.

Parameters:

- `id_or_cachedinfo` - ID or cached information array.

Returns: None

`UsingCache(id)`

Static method to initiate a SmartSearch object using inmemory cache if available. Data will be pulled from the DB if there is no cached data in memory.

Parameters:

- `id` - ID of the SmartSearch.

Returns: SmartSearch object.

`ID()`

Gets the ID of the SmartSearch.

Parameters: None

Returns: Integer ID.

`Exists()`

Checks if the SmartSearch exists.

Parameters: None

Returns: Boolean.

`UpdateValues(params)`

Updates values for the SmartSearch.

Parameters:

- `params` - Associative array of parameters to update.

Returns: None

`Title()`

Gets the title of the SmartSearch.

Parameters: None

Returns: Title string.

`SetTitle(val)`

Sets the title for the SmartSearch.

Parameters:

- `val` - New title string.

Returns: None

`HasTitle()`

Checks if the SmartSearch has a title.

Parameters: None

Returns: Boolean.

`Create()`

Static method creates a new SmartSearch in the database and returns the object.

Returns: SmartSearch object.

`Remove()`

Removes the SmartSearch.

Parameters: None

Returns: None

Company

`CompanyID()`

Gets the company ID associated with the SmartSearch.

Parameters: None

Returns: Integer company ID.

`SetCompanyID(val)`

Sets the company ID for the SmartSearch.

Parameters:

- `val` - New company ID.

Returns: None

`HasCompanyID()`

Checks if the SmartSearch has a company ID.

Parameters: None

Returns: Boolean.

Ownership

`Addedby()`

Gets the ID of the user who added the SmartSearch.

Parameters: None

Returns: Integer user ID.

`SetAddedby(val)`

Sets the user ID who added the SmartSearch.

Parameters:

- `val` - New user ID.

Returns: None

`HasAddedby()`

Checks if the SmartSearch has a user who added it.

Parameters: None

Returns: Boolean.

Search Fields

`FieldsGroupCount()`

Gets the count of field groups in the SmartSearch.

Parameters: None

Returns: Integer count.

`FieldsCount(id_group)`

Gets the count of fields in a specific group of the SmartSearch.

Parameters:

- `id_group` - ID of the field group.

Returns: Integer count.

`isAddButtonHidden(id_group)`

Checks if the "Add" button is hidden for a specific field group in the SmartSearch.

Parameters:

- `id_group` - ID of the field group.

Returns: Boolean.

`Fields()`

Gets the fields information of the SmartSearch.

Parameters: None

Returns: Serialized fields information.

`SetFields(val)`

Sets the fields for the SmartSearch.

Parameters:

- `val` - Serialized fields information.

Returns: None

`HasFields()`

Checks if the SmartSearch has fields information.

Parameters: None

Returns: Boolean.

Usage

`UseCount()`

Gets the use count of the SmartSearch.

Parameters: None

Returns: Integer use count.

`SetUseCount(val)`

Sets the use count for the SmartSearch.

Parameters:

- `val` - New use count.

Returns: None

`HasUseCount()`

Checks if the SmartSearch has a use count.

Parameters: None

Returns: Boolean.

`LastTimeUsed()`

Gets the last time the SmartSearch was used.

Parameters: None

Returns: Integer timestamp.

`SetLastTimeUsed(val)`

Sets the last time the SmartSearch was used.

Parameters:

- `val` - New timestamp.

Returns: None

`HasLastTimeUsed()`

Checks if the SmartSearch has a last time used.

Parameters: None

Returns: Boolean.

Token

`Token()`

Gets the token associated with the SmartSearch.

Parameters: None

Returns: Token string.

`SetToken(val)`

Sets the token for the SmartSearch.

Parameters:

- `val` - New token string.

Returns: None

`HasToken()`

Checks if the SmartSearch has a token.

Parameters: None

Returns: Boolean.

SQL

`SQLOperatorByTag(tagname, value)`

Generates SQL operator based on field tagname and value. You can use them for `WHERE` part of the search.

Parameters:

- `tagname` - Field tag name. Available Tags
- `value` - Value for the field.

Returns: SQL operator string.

`GetSQLQuery()`

Generates the SQL query for the SmartSearch based on its fields.

Parameters: None

Returns: SQL query string.

Timestamps

`Addedtime()`

Gets the time when the SmartSearch was added.

Parameters: None

Returns: Integer timestamp.

`SetAddedtime(val)`

Sets the time when the SmartSearch was added.

Parameters:

- `val` - New timestamp.

Returns: None

`HasAddedtime()`

Checks if the SmartSearch has an added time.

Parameters: None

Returns: Boolean.

Action

`Action()`

Gets the action associated with the SmartSearch.

Parameters: None

Returns: Action string.

`SetAction(val)`

Sets the action for the SmartSearch.

Parameters:

- `val` - New action string.

Returns: None

`HasAction()`

Checks if the SmartSearch has an action.

Parameters: None

Returns: Boolean.

SmartSearchActions class

- Full class name: `GS\SmartSearch\SmartSearchActions`

Information:

- Type: Constant Definitions

This class represents available actions for the Smart Search.

Available Constants

- `CUSTOMERS` - string, Represents the 'customers' smart search category.
- `ORGANIZATIONS` - string, Represents the 'organizations' smart search category.
- `LISTS` - string, Represents the 'lists' smart search category.

`ListAvailable()`

Returns an array of available smart search categories.

Parameters: None

Returns:

- `array` - An array containing available smart search categories.

SmartSearchesList List class

- Full class name: `\GS\SmartSearch\SmartSearchesList`

Information:

- Type: ORM Model List
- DB Table: `smartsearches`
- List Items: \GS\SmartSearch\SmartSearch

`SetFilterCompany($company_or_id)`

Sets a filter for the company in the SmartSearchList.

Parameters:

- `company_or_id` - Company or company ID.

Returns: None

`SetFilterEmployee($employee_or_id)`

Sets a filter for the employee in the SmartSearchList.

Parameters:

- `addedby` - ID of the employee or employee object.

Returns: None

`OrderByAddedTime($asc)`

Orders the SmartSearchList by added time.

Parameters:

- `asc` - Boolean indicating ascending or descending order.

Returns: SmartSearchList object.

PendingSMSMessage class

- Full class name: `\GS\Messages\SMS\SMSMessage`
- Legacy alias (shouldn't be used in new code): ~~`\TMessage`~~

Information:

- Type: ORM Model
- DB Table: `smslog`

This class is designed to work with SMS conversations when repcipients confirm that they want to receive the messages.

This class is descendant of \GS\Messages\SMS\GenericSMS, and it has all parent's methods.

SMSMessageList class

- Full class name: `\GS\Messages\SMS\SMSMessageList`
- Legacy alias (shouldn't be used in a new code): ~~`\TMessageList`~~

Information:

- Type: ORM Model List
- Descendants: `\GS\Messages\SMS\PendingSMSMessageList`
- DB Table: `smslog`
- List items: \GS\Messages\SMS\SMSMessage

This class is designed to work with list of SMS. There is a modification of this class to work with pending messages.

`SetFilterCustomer(customer_or_id)`

Sets a filter for the list to include only items related to the specified customer.

Parameters:

- `customer_or_id` - Int or Object representing the customer.

`SetFilterOrganization(organization_or_id)`

Sets a filter for the list to include only items related to the specified organization.

Parameters:

- `organization_or_id` - Int or Object representing the organization.

`SetFilterCompany(company_or_id)`

Sets a filter for the list to include only items related to the specified company.

Parameters:

- `company_or_id` - Int or Object representing the company.

`SetFilterScheduledTimeBetween(from, to)`

Sets a filter for the list to include only items with scheduled time falling between the specified range.

Parameters:

- `from` - Integer, start of the range.
- `to` - Integer, end of the range.

`SetFilterAddedTimeBetween(from, to)`

Sets a filter for the list to include only items added between the specified time range.

Parameters:

- `from` - Integer, start of the range.
- `to` - Integer, end of the range.

`SetFilterReferral(referral_or_id)`

Sets a filter for the list to include only items related to the specified referral.

Parameters:

- `referral_or_id` - Int or Object representing the referral.

SetFilterMarketingMessages()

Sets a filter for the list to include only marketing-related messages such as 'multi', 'blast', and 'notification'.

SetFilterIncoming()

Sets a filter for the list to include only incoming items.

SetFilterOutgoing()

Sets a filter for the list to include only outgoing items.