



TRIPOLIS SOLUTIONS

TripolisDialogue



Webservices 3.2

Tripolis Solutions

Tripolis Dialogue Webservices



1.	Introduction.....	5
2.	WSDL	6
3.	Types.....	7
3.1.	AuthInfo.....	7
3.2.	ContactInfo	7
3.3.	ContactWithGroupMembershipsInfo	7
3.4.	smartGroupInfo.....	7
3.5.	fieldCondition.....	8
3.6.	contactGroupCondition	8
3.7.	DatabaseField	8
3.8.	DatabaseGroup	9
3.9.	GroupMembershipInfo.....	9
3.10.	KeyValuePair.....	9
3.11.	SearchParameter	9
3.12.	TimeRange	10
3.13.	Mailing	10
3.14.	Fault	11
3.15.	ResponseCode	11
3.16.	TellAFriend	11
3.17.	MailingSnapshot	11
3.18.	ClickedUrl	11
3.19.	OpenInfo	12
3.20.	ApiResponse	12
3.21.	ArticleInfo	12
3.22.	Attachment.....	13
3.23.	BouncedContact	13
3.24.	ContactDatabaseInfo	14
3.25.	smartGroupInfo.....	14
3.26.	SmsMessage.....	14
3.27.	DirectEmail	14
3.28.	DirectEmailType	15
3.29.	ImageInfo	15
3.30.	ImportStatus	15
3.31.	MailingSummaryReport	16
3.32.	NewsletterInfo	16
3.33.	NewsletterPreview	17
3.34.	NewsletterSectionInfo	17
3.35.	NewsletterType	17
3.36.	NotSendContact	18
3.37.	NotSendContacts.....	18
3.38.	WorkspaceInfo	18
4.	Methods in contact management module.....	19
4.1.	upsertContact	19
4.2.	upsertContactWithNewsletterResponse.....	20
4.3.	getContacts	22
4.4.	searchContacts	22
4.5.	getContactsWithGroupMemberships	23
4.6.	removeContactGroupMembership	23
4.7.	getDatabaseFields	24
4.8.	getDatabaseGroups	25

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

5.	Advanced API calls	27
5.1.	exportContacts.....	27
5.2.	exportClientMailBlacklist.....	27
5.3.	exportContactDatabaseMailBlacklist	28
5.4.	createContactGroup	29
5.5.	upsertSmartgroup	30
5.6.	deleteSmartgroups	31
5.7.	getSmartgroups	32
5.8.	importContacts	33
5.9.	getImportStatus.....	36
5.10.	createDatabase	37
5.11.	createDatabaseField.....	37
5.12.	deleteDatabaseField.....	37
5.13.	assignContactsToGroup	38
5.14.	getMailings	39
5.15.	getMailJobs.....	41
5.16.	getSmsJobs	42
5.17.	getMailing.....	44
5.18.	getLastMailingSnapshot.....	45
5.19.	getMailingSnapshot.....	47
5.20.	tellAFriend	48
5.21.	previewDirectEmail	50
5.22.	scheduleMailing	52
5.23.	scheduleSms.....	53
5.24.	updateMailingStatus	55
5.25.	getDeliveredByMailing	55
5.26.	getDeliveredBySmsJob	57
5.27.	getNotSendByMailing	57
5.28.	getNotSentByMailing.....	57
5.29.	getNotSentBySmsJob	59
5.30.	getClicksByMailing	59
5.31.	getOpensByMailing	61
5.32.	getCommunicationHistoryByContact	62
5.33.	exportDeliveredByMailing	62
5.34.	getExportFile	63
5.35.	assignArticleToNewsletter	65
5.36.	deleteContact.....	66
5.37.	deleteContactGroup	66
5.38.	getMaxValueOfContactField.....	67
5.39.	getArticle.....	67
5.40.	getBouncesByMailing	68
5.41.	getBouncedBySmsJob	69
5.42.	getContactDatabases	69
5.43.	getDirectEmailTypes	70
5.44.	getDirectEmails	71
5.45.	getNewsletterSections.....	72
5.46.	getNewsletterTypes	72
5.47.	getNewsletters	73
5.48.	getSmsMessageTypes	74
5.49.	getSummaryReport	75

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

5.50.	upsertWorkspace.....	76
5.51.	getWorkspaces.....	77
5.52.	deleteWorkspace.....	77
5.53.	previewNewsletter.....	78
5.54.	upsertDirectEmailType.....	79
5.55.	upsertNewsletterType.....	81
5.56.	upsertSmsMessageType.....	82
5.57.	deleteDirectEmailType.....	83
5.58.	deleteNewsletterType.....	84
5.59.	deleteSmsMessageType.....	85
5.60.	deleteSmsMessage.....	86
5.61.	upsertArticle.....	87
5.62.	upsertImage.....	88
5.63.	upsertDirectMail.....	89
5.64.	upsertNewsletter.....	91
5.65.	upsertSmsMessage.....	92

Tripolis Solutions

Tripolis Dialogue Webservices



1. INTRODUCTION

The Tripolis Dialogue SOAP Application Programming Interface (API) enables you to integrate Dialogue directly in other tools and applications. Tripolis uses the SOAP and WSDL standards so you can develop in your own favorite environment such as .Net, Java, Perl or PHP.

The Tripolis Dialogue SOAP API is an extensive web service with various possibilities. This web service allows someone to call from a distance using a client (for example a web browser) to a service or fire a command to the Tripolis Dialogue Platform.

Samples are: to add a person to a database (subscribe) or to unsubscribe someone from a certain mailing. But also the execution of a mail campaign is a possibility.

See also: http://en.wikipedia.org/wiki/Web_service For a more precise description.

This underlying document describes the various functions (methods) the API consists.

To use those functions standard SOAP calls will be used. Various programming languages (PHP, ASP.net, java, etc..) have standard build in support for SOAP.

Various articles on the internet can be helpful with the use of the API.

See also: <http://en.wikipedia.org/wiki/SOAP>

For use of the Tripolis Dialogue web service by means of SOAP a license is necessary. The web service licenses are activated by the Tripolis System Administrators.

2. WSDL

The Web Services Description Language (WSDL, pronounced 'wiz-dəl' or spelled out, 'W-S-D-L') is an XML-based language that provides a model for describing Web services.

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

(Source: http://en.wikipedia.org/wiki/Web_Services_Description_Language)

Please contact Tripolis Support for the correct WSDL for your Tripolis Dialogue instance.

This web service is MTOM enabled.

MTOM stands for SOAP "Message Transmission Optimization Mechanism" and it is a W3C standard. It uses binary optimization (Mime), which makes it possible to send larger attachments.

No-mtom

If you don't want to use binary optimization (Mime) when sending (small) attachments you can better use the no-mtom service.

Please contact Tripolis Support for the correct WSDL for your Tripolis Dialogue instance.

Some clients (php/.net) will find it easier to communicate with webservice without MTOM.

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

3. TYPES

3.1. AuthInfo

Holds the credentials of the user that wants to use the api call. This object is needed for every call that is made to the service.

Name	Type	Description
client	string	reference to client domain in TripolisDialogue
username	string	user with rights to log into the environment
password	string	password for this user

3.2. ContactInfo

Represents a contact in a database. The contactId is encrypted and can be used when trying to update or retrieve the contact. The keyValuePairs holds the names of the database fields and their values for the contact. The structure - which types - can diver depending on the structure of the database; a contact in one database can have a totally different structure than a contact in another database.

Name	Type	Description
contactId	string	Encrypted contactId
keyvaluepairs	List<KeyValuePair>	List of KeyValuePairs.
keyValuePairsMap	Map	Keyvaluepairs in a map

3.3. ContactWithGroupMembershipsInfo

Represents a contact including the group membership of the contact.

Name	Type	Description
contactInfo	ContactInfo	ContactInfo of the contact
groupMembershipInfos	List<GroupMembershipInfo>	Group membership info of the contact

3.4. smartGroupInfo

Represents a smartgroup.

Name	Type	Description
label	String	Label of the smartgroup
name	String	Name of the smartgroup
queryCondition	fieldCondition or contactGroupCondition	Either a fieldCondition or a contactGroupCondition
queryType	String	Type of query: ANY ALL ADVANCED. Defaults to ANY
advancedString	String	Advanced query e.g. (1 AND 2) OR 3
status	String	Smartgroup status : ACTIVE

		ARCHIVED Defaults to ACTIVE
--	--	--------------------------------

3.5. fieldCondition

Represents a contact field condition for use with queryCondition

Name	Type	Description
fieldName	String	Name of the contact field
operator	String	Operator used for the query Available operators: EQUALS NOT EQUALS CONTAINS DOES NOT CONTAIN STARTS WITH ENDS WITH LESS THAN LESS THAN OR EQUAL GREATER THAN GREATER THAN OR EQUAL IS SET IS NOT SET
value	String	Value to be used in combination with the operator
orderBy	Boolean	Not implemented yet
sortOrder	String	Not implemented yet
searchStringValue	Boolean	Search fields as string type

3.6. contactGroupCondition

Represents a contact group condition for use with queryCondition

Name	Type	Description
groupName	String	The name of the group
inGroup	Boolean	Specifies if the contacts must be in or not in the specified contactgroup. Defaults to false
includeSubgroups	Boolean	Specifies if subgroups of the specified contactgroup should be included. Defaults to false

3.7. DatabaseField

Complex representing a contact database field:

Name	Type	Description
name	string	Name of the field
label	string	Label of the field
type	string	One of those uppercase values [STRING, INT, DECIMAL, DATE, DATETIME, BOOLEAN, MOBILE, EMAIL, PICKLIST]

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

defaultValue	string	Default value of the field
key	boolean	Is the field a unique key field?
mandatory	boolean	Is the field required?
minimumLength	integer	Minimum length of the field values
maximumLength	integer	Maximum length of the field values
picklistItems	List<KeyValuePair>	If the field has the type picklist, then the picklist items are represented in a list of KeyValuePair.

3.8. DatabaseGroup

Complex representing a contact database group:

Name	Type	Description
name	String	Name of the group
label	String	Label of the group
type	String	The type of the group, one of the following: <ul style="list-style-type: none"> - STATIC - TEST - SUBSCRIPTION
parentGroupName	String	The name of the parent group.

3.9. GroupMembershipInfo

Holds information about the groups to which contact belong.

Name	Type	Description
groupLabel	string	Label of the contact group
groupName	string	Name of the contact group
subscribed	boolean	Subscribe status of the contact
confirmed	boolean	Confirmed status of the contact

NOTE: the groupMembershipInfo complex type, has the 2 parameters subscribed and confirmed. These parameters are used for a double OPT-IN subscribe/change of a contact; contact which have the status subscribed=true and the confirmed=false can NOT be emailed from within the application/mailer. Only contacts with both parameters true can be emailed. Of-course the double OPT-IN email (provided in the emailname parameter) will be send to the confirmation email.

3.10. KeyValuePair

KeyValuePair are used to represent field values.

Name	Type	Description
key	string	Name of the field
value	string	Value of the field

3.11. SearchParameter

Represents a search parameter. Like the KeyValuePair object but with the addition of the 'operator' field which defines how the search query will be built up.

Name	Type	Description
key	string	Search field
value	string	Search value

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

operator	string	Operator for the search. Possible values are: EQUALS STARTS_WITH ENDS_WITH CONTAINS LESS_THAN LESS_THAN_OR_EQUAL GREATER_THAN GREATER_THAN_OR_EQUAL NOT_EQUALS IN NOT_IN DOES_NOT_CONTAIN
----------	--------	---

3.12. TimeRange

Complex type to represent a time range with start and end date.

Name	Type	Description
startDate	xsd:dateTime	Start date
endDate	xsd:dateTime	End date

3.13. Mailing

Representation of a mailjob in Tripolis Dialogue.

Name	Type	Description
mailingId	string	Encrypted id of the mailing
mailingName	string	Name of the mailing (consists of subject and formatted startTime)
subject	string	Subject of the mailing
startTime	xsd:dateTime	Start time (schedule time) of the mailing
endTime	xsd:dateTime	End time of the mailing
campaignName	string	Name of the campaign if the mailing was sent by a campaign; can be empty
databaseName	string	Name of the contact database
groupName	string	Name of the contact group; can be empty
smartgroupName	string	Name of the smartgroup; can be empty
mailsPerHour	integer	Number of mails sent per hour
mailInterval	integer	
sampleNumber	integer	Number of sample mails sent
status	string	Current status of mailing
error	string	Reason of failure in case of failure; can be empty
isRemoved	boolean	
createdBy	string	Username of user that scheduled the mailing
requestedNumberOfSend	integer	Number of contacts that should have received the mailing
numberOfSend	integer	Number of contacts that actually received the mailing
numberOfSkipped	integer	Number of contacts that were skipped

Tripolis Solutions

Tripolis Dialogue Webservices



3.14. Fault

Type to represent the returned fault.

Name	Type	Description
faultcode	string	Standard fault code
faultstring	string	Description of the fault
detail	exception	Can be an InvalidInputException, InvalidAuthInfoException or TripolisUnknownException

3.15. ResponseCode

Type to represent the returned response.

Name	Type	Description
responseCode	string	Standard response code
responseDescription	string	Description of the response
message	string	Message of the response

3.16. TellAFriend

Complex representing the receiver of a tell-a-friend mail snapshot.

Name	Type	Description
email	string	Email address of the receiver
profileFields	List<KeyValuePair>	ProfileFields which are represented in a list of KeyValuePair.

3.17. MailingSnapshot

Extension of the Mailing object with added html and text content.

Name	Type	Description
<i>Fields of Mailing object</i>		
html	string	Parsed html content as it was sent
text	string	Parsed text content as it was sent

3.18. ClickedUrl

Type to represent a clicked url.

Name	Type	Description
<i>Fields of Mailing object</i>		
Browser	string	Browser of the contact that clicked
contactId	string	Encrypted ID of the contact
ipAddress	String	IP address of the contact that clicked

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

jobId	String	Encrypted ID of the mailing/job
OS	String	Operating system of the contact that clicked
timeStamp	String	The time the contact clicked
url	String	Url where the contact clicked on

3.19. OpenInfo

Type to represent an open.

Name	Type	Description
<i>Fields of Mailing object</i>		
contactId	string	Encrypted ID of the contact
jobId	string	Encrypted ID of the mailing/job
ip	string	IP address of the contact that clicked
browser	string	Browser of the contact that clicked
timeStamp	string	The time the contact clicked

3.20. ApiResponse

This object represents a number of responses that can be recognized and matched by the responsecode. At this moment the available response codes that might be received are:

100 : success

There will be other codes added in the future.

Name	Type	Description
responseCode	string	The numeric response code
responseDescription	string	The explanation of the response code
Message	string	A custom field that can be used to elaborate on the response.

3.21. ArticleInfo

Represents a Article in a database. An Article object is a mix of article fields and their values and a rendered article represented as a String.

Name	Type	Description
articleId	string	The encrypted id of the article
label	string	The label of the article
name	string	The name of the article
articleTypeDefinitionName	string	The name of the ArticleTypeDefinition object
modifiedBy	string	The username of the person that last modified the article
modifiedDate	xsd:dateTime	The dat of the last modification
tags	List<String>	A list of tags representing the

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

		MediaCategories that are attached to the article
articleFields	List<KeyValuePair>	A list of keyvaluepairs representing the article fields. The key is the name of the field descriptor.

3.22. Attachment

Wrapper for an attachment to be used in service calls. This can be any type of file; the type will be set to application/octet-stream.

Name	Type	Description
fileName	string	The name of the file
data	xs:base64Binary	The DataHandler that will contain the actual data of the file. The data can be read by retrieving a byte array from the DataHandler object.

3.23. BouncedContact

Represents a contact that is registered as a bounce for one specific email address.

Name	Type	Description
contactInfo	ContactInfo	The contact information of the contact that has a bounced emailaddress
emailAddress	String	The emailaddress that caused the bounce
receivedBounce	xs:dateTime	The time the bounce was registered
bounceCode	string	The bounce code
bounceCategoryId	int	The bounce category
bounceCategoryeDescription	string	The description of the bounce category
bounceReason	string	The reason of the bounce

bounceCategoryId	Description	Hard Bounce	Parent Id
1	bounce.category.invalid.user	Y	
2	bounce.category.user.unknown	Y	1
3	bounce.category.syntax.error	Y	1
4	bounce.category.account.disabled	Y	1
5	bounce.category.invalid.domain	Y	
6	bounce.category.unknown.domain	Y	5
8	bounce.category.routing.errors	N	
9	bounce.category.no.answer.from.host	N	8
10	bounce.category.relaying.denied	N	8
11	bounce.category.unable.to.route	N	8
12	bounce.category.blocking.errors	N	
13	bounce.category.policy.issues	N	12
14	bounce.category.blacklist.issues	N	12

15	bounce.category.content.issues	N	12
16	bounce.category.mailbox.related	N	
17	bounce.category.temporarily.disabled	N	16
18	bounce.category.quota.exceeded	N	16
19	bounce.category.other	N	
20	bounce.category.timeout	N	19
21	bounce.category.osi	N	19
22	bounce.caterogy.mailbox.status	N	19
0	bounce.category.unknown	N	19

3.24. ContactDatabaseInfo

Small representation of a ContactDatabase.

Name	Type	Description
name	string	The name of the contact database
label	string	The label of the contact database
contactDatabaseId	string	The encrypted id of the contact database

3.25. smartGroupInfo

Representation of a smartgroup.

Name	Type	Description
name	string	The name of the contact database
label	string	The label of the contact database
queryCondition	string	The encrypted id of the contact database
contact		

3.26. SmsMessage

Represents a Sms message.

Name	Type	Description
Id	string	The encrypted id of the sms message
label	string	the label (required)
name	string	The name (required)
description	string	A description (optional)
originator	string	The name of the sender of the sms (optional)
message	string	The content of the sms (required)
alternativeMessage	string	The alternative sms content (required)

3.27. DirectEmail

Represents a DirectEmail.

Name	Type	Description
directEmailId	string	The encrypted id of the email (required for

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

		updates)
label	string	the label (required)
name	string	The name (required)
subject	string	The subject (required)
description	string	A description (optional)
fromName	string	The name of the sender of the email (optional)
fromAddress	string	The email address of the sender of the email (optional)
replyTo	string	The email address used when replying to the email (optional)
html	string	The html content (optional)
text	string	The text content (optional)

3.28. DirectEmailType

Represents a DirectEmail type definition object.

Name	Type	Description
label	string	The label
name	string	The name
fromName	string	The default name of the sender
fromAddress	string	The default email address of the sender
replyTo	string	The default email address to reply to
htmlUrl	string	The url for retrieval of external html content
textUrl	string	The url for retrieval of external text content
encoding	string	The encoding of the content
emailFieldName	string	The name of the email field in the database that is used to send to
editorEnabled	boolean	True if system html editor is enabled; otherwise false

3.29. ImageInfo

Representation of an image inside the system. Used for e.g. uploading images into the system.

Name	Type	Description
imageId	string	The encrypted id of the image (required for updates)
attachment	Attachment	The attachment holding the data of the image
name	string	The name of the image
label	string	The label of the image
description	string	The description of the image
tags	List<string>	Optional tags to enable easy searching inside the system.

3.30. ImportStatus

Represents the status of an import.

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

Name	Type	Description
running	boolean	True if the import is still running; otherwise false
startTime	xs:dateTime	The date and time the import started
endTime	xs:dateTime	The date and time the imported ended (or updated the status)
importMode	string	The import mode (ADD_OR_UPDATE, ADD_TO_GROUP, EXCLUDE_FROM_GROUP, ONLY_ALLOW_ADD, ONLY_ALLOW_ADD_TO_GROUP, ONLY_ALLOW_UPDATE, SYNCHRONIZE_GROUP)
created	int	Number of contact that were created
updated	int	Number of contacts that were updated
addedToGroup	int	Number of contacts added to a group
removedFromGroup	int	Number of contacts removed from a group
numberOfErrors	int	Number of erroneous contacts
importedBy	string	Username of user that started the import
importErrors	List<string>	A list of errors

3.31. MailingSummaryReport

Representation of the statistics of a mailing.

Name	Type	Description
<i>Fields of Mailing object</i>		
queued	int	Number of contacts queued
queuedPct	double	Percentage of contacts queued
hardBounces	int	Number of hard bounces
hardBouncesPct	double	Percentage of hard bounces
softBounces	int	Number of soft bounces
softBouncesPct	double	Percentage of soft bounces
totalOpens	int	Number of total opens
uniqueOpens	int	Number of unique opens
totalClicks	int	Number of total clicks
uniqueClicks	int	Number of unique clicks
ctrDelivered	int	Number of delivered 'Click Through Rate'
ctrOpened	int	Number of opened 'Click Through Rate'
replyTo	string	Email address used for replies
fromAddress	string	The email address of the sender
fromName	string	The name of the sender
mailingType	string	The type of the mailing (either mail, sms or campaign)
typeName	string	The name of the type definition

3.32. NewsletterInfo

Representation of a Newsletter object.

Name	Type	Description
newsletterId	string	The encrypted id of the newsletter
label	string	The label
name	string	The name
subject	string	The subject
fromAddress	string	The email address of the sender
fromName	string	The name of the sender
newsletterTypeName	string	The name of the type definition
modifieddate	xs:dateTime	The date of the last modification

3.33. NewsletterPreview

Represents the newsletter including the content.

Name	Type	Description
<i>Fields of NewsletterInfo object</i>		
htmlContent	string	The html content of the newsletter
textContent	string	The text content of the newsletter

3.34. NewsletterSectionInfo

Representation of a newsletter section.

Name	Type	Description
newsletterSectionId	string	The encrypted id of the section
label	string	The label
name	string	The name
rssUrl	string	The url of the rss feed to fill the section
numberOfRssArticles	int	The number of articles to be shown
rssActive	boolean	True if the feed is active; otherwise false

3.35. NewsletterType

Represents a Newsletter type on which newsletters are based.

Name	Type	Description
label	string	The label
name	string	The name
fromName	string	The default name of the sender
fromAddress	string	The default email address of the sender
replyTo	string	The default email address to reply to
encoding	string	The encoding of the content
emailFieldName	string	The name of the email field in the contact database
defaultSubject	string	The default subject to be used

3.36. NotSendContact

Represents a contact that has not received a particular mailing.

Name	Type	Description
contactInfo	ContactInfo	The contact information
notSentReason	string	The reason why the contact did not receive the mailing

3.37. NotSendContacts

Representation of all contacts that did not receive a particular mailing.

Name	Type	Description
notSentContacts	List< NotSendContact >	A list of contacts that did not receive the mailing
jobId	string	The encrypted mailing id

3.38. WorkspaceInfo

Representation of a workspace.

Name	Type	Description
label	string	The label
name	string	The name
workspaceId	string	The encrypted workspace id
contactDatabaseId	string	The encrypted contact database id

4. METHODS IN CONTACT MANAGEMENT MODULE

4.1. upsertContact

Creates and/or updates a contact. Can also handle subscription to one or more groups. When creating a contact the contactInfo only needs to have the required fields that have no default value. After successful creation of the contact the response will contain an encrypted contactId that can be used for all other actions on this contact. When updating a contact the contactInfo only needs the values that need to be changed. In these cases it is not needed to provide any groupMemberships unless a subscription needs to be updated also e.g. an unconfirmed subscription needs to be confirmed. The only thing that is required is the encrypted contactId that was given at creation time of the contact. This contactId is also provided when a contact is retrieved with one of the other api calls.

This method foresees in the complete subscription flow: subscribe without confirmation, confirm subscription (double opt-in) and unsubscribe from group. This is done by setting the 2 booleans in the groupMembershipInfo objects: subscribed and confirmed

There are 3 possible settings for a subscription:

1. subscribed=true / confirmed=false : the contact will be subscribed but is not yet confirmed. This person will be available in the group but will not receive any mails sent to this group until confirmation has occurred.
2. subscribed=true / confirmed=true : the contact will be confirmed. From now on the contact is a full member of the group and will receive mails sent to the group
3. subscribed=false : the contact will be unsubscribed from the group (the confirmed field is irrelevant in this case).

Note: When assigning a contact to a group the groupLabel is ignored. This value will be filled when using the getContactsWithGroupMemberships call.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
groupMembership	GroupMembershipInfo[]	(optional) holds information about the groups to which the contact will be subscribed.
contactInfo	ContactInfo	(optional) holds the fields which will define the contact. NOTE: this object may contain optional as well as required fields
workspaceName	String	(optional) name of the workspace
emailName	String	(optional) name of the email
externalResponseHttp	String	(optional) address of an external http server which then will receive the response of the call
refName	String	referrer name of the client
refIp	String	referrer ip of the client

Tripolis Solutions

Tripolis Dialogue Webservices



Returns:

The encrypted contact id if all went well, which might be appended with an extra message e.g. when the responseUrl could not be reached correctly. This message will start with a semi-colon. If something went wrong an exception will be thrown.

4.2. upsertContactWithNewsletterResponse

Creates and/or updates a contact. Can also handle subscription to one or more groups. When creating a contact the contactInfo only needs to have the required fields that have no default value. After successful creation of the contact the response will contain an encrypted contactId that can be used for all other actions on this contact. When updating a contact the contactInfo only needs the values that need to be changed. In these cases it is not needed to provide any groupMemberships unless a subscription needs to be updated also e.g. an unconfirmed subscription needs to be confirmed. The only thing that is required is the encrypted contactId that was given at creation time of the contact. This contactId is also provided when a contact is retrieved with one of the other api calls.

This method foresees in the complete subscription flow: subscribe without confirmation, confirm subscription (double opt-in) and unsubscribe from group.

This is done by setting the 2 booleans in the groupMembershipInfo objects: subscribed and confirmed

There are 3 possible settings for a subscription:

1. subscribed=true / confirmed=false : the contact will be subscribed but is not yet confirmed. This person will be available in the group but will not receive any mails sent to this group until confirmation has occurred.
2. subscribed=true / confirmed=true : the contact will be confirmed. From now on the contact is a full member of the group and will receive mails sent to the group
3. subscribed=false : the contact will be unsubscribed from the group (the confirmed field is irrelevant in this case).

Note: When assigning a contact to a group the groupLabel is ignored. This value will be filled when using the getContactsWithGroupMemberships call.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
groupMembership	GroupMembershipInfo[]	(optional) holds information about the groups to which the contact will be subscribed.
contactInfo	ContactInfo	(optional) holds the fields which will define the contact. NOTE: this object may contain optional as well as required fields
workspaceName	String	(optional) name of the workspace
newsletterName	String	(optional) name of the newsletter
externalResponseHttp	String	(optional) address of an external http

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

		server which then will receive the response of the call
refName	String	referrer name of the client
refIp	String	referrer ip of the client

Returns:

The encrypted contact id if all went well, which might be appended with an extra message e.g. when the responseUrl could not be reached correctly. This message will start with a semi-colon. If something went wrong an exception will be thrown.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertContactWithNewsletterResponse>
      <authInfo>
        <client>client</client>
        <username>user</username>
        <password>password</password>
      </authInfo>
      <databaseName>databasename</databaseName>
      <groupMembership>
        <groupLabel></groupLabel>
        <groupName>group</groupName>
        <subscribed>>true</subscribed>
        <confirmed>>false</confirmed>
      </groupMembership>
      <contactInfo>
        <contactId></contactId>
        <keyValuePairs>
          <key>email</key>
          <value>email@email.com</value>
        </keyValuePairs>
        <keyValuePairs>
          <key>name</key>
          <value>Naam</value>
        </keyValuePairs>
      </contactInfo>
      <workspaceName>workspace</workspaceName>
      <newsletterName>newsletterName</newsletterName>
      <externalResponseHttp></externalResponseHttp>
      <refName>localhost</refName>
      <refIp>192.168.25.40</refIp>
    </api:upsertContactWithNewsletterResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.4</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:upsertContactWithNewsletterResponseResponse xmlns:ns1="http://api.tripolis.com/">
      <response>MTQ3MTQ3MTSnf73cr*wzow</response>
    </ns1:upsertContactWithNewsletterResponseResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



4.3. getContacts

Retrieves an array of contacts by one or more search parameters. The values of these parameters need to be complete because the query will always use the 'EQUALS' query operator.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
contactId	String	encrypted id of the contact for direct retrieval (optional; if available it will precede getParameter)
getParameter	KeyValuePair[]	getParameter (optional; only used in case of absence of contactId)

Returns:

Array of ContactInfo objects

4.4. searchContacts

Retrieves an array of contacts by one or more search parameters. The values of these parameters can be partial.

An example SearchParameter to search for all emails containing the word 'hotmail' would be:

key = email
value = hotmail
operator = CONTAINS

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
searchParameter	SearchParameter[]	the searchParameters

Returns:

Array of ContactInfo objects

Tripolis Solutions

Tripolis Dialogue Webservices



4.5. getContactsWithGroupMemberships

Retrieves an array of contacts by one or more search parameters. The values of these parameters need to be complete because the query will always use the 'EQUALS' query operator.

In addition to the contactInfo objects there will be an array of groupMembershipInfo objects available for every contactInfo specifying to which group the contact belongs.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
contactId	String	encrypted id of the contact for direct retrieval (optional; if available it will precede getParameter)
getParameter	KeyValuePair[]	getParameter (optional; only used in case of absence of contactId)

Returns:

Array of groupMembershipInfo objects.

4.6. removeContactGroupMembership

Removes one or more group assignments for the specified contact.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
contactId	String	encrypted id of the contact
groupMembership	List	List of groups (Strings)
refName	String	referrer name of the client
refIp	String	referrer ip of the client

Returns:

A String containing the message 'membership of group " is removed' appended with a new line ('\n') and so on for every subscription that is successfully removed.

Tripolis Solutions

Tripolis Dialogue Webservices



4.7. getDatabaseFields

Retrieves the fields of the specified database with all the properties. When the field is of type picklist the items will be provided in an array inside the picklist field. This method is useful when building dynamic subscription procedures for instance.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database

Returns:

Array of DatabaseField objects

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getDatabaseFields>
      <authInfo>
        <client>my company</client>
        <username>soap_user@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <databaseName>Mydatabase</databaseName>
    </api:getDatabaseFields>
  </soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:getDatabaseFieldsResponse xmlns:ns1="http://api.tripolis.com/">
      <databaseField>
        <label>Birthday</label>
        <name>birthday</name>
        <defaultValue/>
        <type>date</type>
        <key>>false</key>
        <mandatory>>false</mandatory>
        <minimumLength>1</minimumLength>
        <maxLength>40</maxLength>
      </databaseField>
      <databaseField>
        <label>Name</label>
        <name>name</name>
        <defaultValue/>
        <type>string</type>
        <key>>true</key>
        <mandatory>>true</mandatory>
        <minimumLength>1</minimumLength>
        <maxLength>40</maxLength>
      </databaseField>
    </ns1:getDatabaseFieldsResponse>
  </soap:Body>
</soap:Envelope>
```



```
<label>email</label>
<name>email</name>
<defaultValue/>
<type>email</type>
<key>true</key>
<mandatory>true</mandatory>
<minimumLength>1</minimumLength>
<maxLength>255</maxLength>
</databaseField>
<databaseField>
<label>mobile phone</label>
<name>mobilephone</name>
<defaultValue/>
<type>mobile</type>
<key>false</key>
<mandatory>true</mandatory>
<minimumLength>0</minimumLength>
<maxLength>0</maxLength>
</databaseField>
<databaseField>
<label>Country</label>
<name>country</name>
<defaultValue/>
<type>picklist</type>
<key>true</key>
<mandatory>true</mandatory>
<minimumLength>1</minimumLength>
<maxLength>4</maxLength>
<picklistItems>
<key>Holland</key>
<value>NED</value>
</picklistItems>
<picklistItems>
<key>France</key>
<value>FRA</value>
</picklistItems>
<picklistItems>
<key>Germany</key>
<value>GER</value>
</picklistItems>
<picklistItems>
<key>Italy</key>
<value>ITA</value>
</picklistItems>
</databaseField>
</ns1:getDatabaseFieldsResponse>
</soap:Body>
</soap:Envelope>
```

4.8. getDatabaseGroups

Retrieves the groups of the specified database. The response gives the following:

- an array of databaseGroup object with a name, label, type of group and parent group name

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database

Tripolis Solutions

Tripolis Dialogue Webservices



Returns:

Array of databaseGroups

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getDatabaseGroups>
      <authInfo>
        <client>my company</client>
        <username>soap_user@tripolis.com</username>
        <password>***</password>
      </authInfo>
      <databaseName>Mydatabase</databaseName>
    </api:getDatabaseGroups>
  </soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:getDatabaseGroupsResponse xmlns:ns1="http://api.tripolis.com/">
      <databaseGroup>
        <label>My Group</label>
        <name>mygroup</name>
        <type>STATIC</type>
        <parentGroupName/>
      </databaseGroup>
      <databaseGroup>
        <label>My subgroup</label>
        <name>mysubgroup</name>
        <type>STATIC</type>
        <parentGroupName>mygroup<parentGroupName/>
      </databaseGroup>
      <databaseGroup>
        <label>Test group</label>
        <name>TestGroup</name>
        <type>TEST</type>
        <parentGroupName/>
      </databaseGroup>
    </ns1:getDatabaseGroupsResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



5. ADVANCED API CALLS

5.1. exportContacts

Exports the contacts of the specified database or contact group with specified fieldnames if provided. Otherwise all fields will be exported. The call only returns a CSV attachment.

Parameters:

Name	Type	Description
authInfo	AuthInfo	object used for authentication of user
databaseName	String	name of the database
groupName	String	Name of the contact group
databaseFieldName	List	List of databasefield names

Returns:

Attachment

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:exportContacts>
      <authInfo>
        <client>my company</client>
        <username>soap_user@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <databaseName>TestDatabase</databaseName>
      <groupName></groupName>
      <databaseFieldName>name</databaseFieldName>
    </api:exportContacts>
  </soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:exportContactsResponse xmlns:ns1="http://api.tripolis.com/">
      <exportContacts>
        <fileName>TestDatabase_18-02-2008.csv</fileName>
        <data>
          <xop:Include href="cid:04aef644-554f-4875-85b1-
e1f2faedcdce@http%3A%2F%2Fcxf.apache.org%2F" xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
        </data>
      </exportContacts>
    </ns1:exportContactsResponse>
  </soap:Body>
</soap:Envelope>
```

5.2. exportClientMailBlacklist

Exports the client wide email blacklist.

Tripolis Solutions

Tripolis Dialogue Webservices



Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
timeRange	TimeRange	Start and end date (optional)

Returns:

A csv attachment with blacklisted addresses.

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:exportClientMailBlacklist>
      <authInfo>
        <client>my company</client>
        <username>soap_manual@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <timeRange>
        <startDate>2008-02-26T13:02:36.015+01:00</startDate>
        <endDate>2008-02-27T13:02:36.015+01:00</endDate>
      </timeRange>
    </api:exportClientMailBlacklist>
  </soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:exportClientMailBlacklistResponse xmlns:ns1="http://api.tripolis.com/">
      <response>
        <fileName>MyCompany_blacklist_2008-02-27.csv</fileName>
        <data>
          <xop:Include href="cid:b96b1e6d-2f4b-4ac6-873d-
4348f1317ced@http%3A%2F%2Fcf.apache.org%2F"
xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
        </data>
      </response>
    </ns1:exportClientMailBlacklistResponse>
  </soap:Body>
</soap:Envelope>
```

5.3. exportContactDatabaseMailBlacklist

Exports the mail blacklist for a specified contact database, including database fields and contact ids.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database
timeRange	TimeRange	Start and end date (optional)

Tripolis Solutions

Tripolis Dialogue Webservices



Returns:

A csv attachment with blacklisted contacts including contact data.

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:exportContactDatabaseMailBlacklist>
      <authInfo>
        <client>my company</client>
        <username>soap_manual@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <database>my_database</database>
      <timeRange>
        <startDate>2008-02-22</startDate>
        <endDate>2008-02-24</endDate>
      </timeRange>
    </api:exportContactDatabaseMailBlacklist>
  </soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:exportContactDatabaseMailBlacklistResponse xmlns:ns1="http://api.tripolis.com/">
      <response>
        <fileName>my_database_blacklist_2008-02-28.csv</fileName>
        <data>
          <xop:Include href="cid:bd0017da-ee75-4627-87d4-
2ee76aa01d45@http%3A%2F%2Fcf.apache.org%2F"
xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
        </data>
      </response>
    </ns1:exportContactDatabaseMailBlacklistResponse>
  </soap:Body>
</soap:Envelope>
```

5.4. createContactGroup

Creates a contact group in the specified database. If no group type is provided a STATIC group will be created.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database
contactGroup	ContactGroup	Properties of the new contact group

Returns:

String with success message.

Tripolis Solutions

Tripolis Dialogue Webservices



Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:createContactGroup>
      <authInfo>
        <client>my company</client>
        <username>soap_user@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <databaseName>Mydatabase</databaseName>
      <contactGroup>
        <label>sOAP_group</label>
        <name>SOAP Group</name>
        <type>STATIC</type>
        <parentGroupName></parentGroupName>
      </contactGroup>
    </api:createContactGroup>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:createContactGroupResponse xmlns:ns1="http://api.tripolis.com/">
      <response>ContactGroup 'SOAPGroup' succesfully created.</response>
    </ns1:createContactGroupResponse>
  </soap:Body>
</soap:Envelope>
```

5.5. upsertSmartgroup

Creates or updates a smartgroup in the specified database.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database
smartGroupInfo	smartGroupInfo	smartGroupInfo object

Returns:

Api response object

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertSmartGroup>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
    </api:upsertSmartGroup>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</authInfo>
<databaseName>default</databaseName>
<smartGroupInfo>
  <label>smartgroup</label>
  <queryCondition>
    <fieldCondition>
      <fieldName>firstname</fieldName>
      <operator>IS SET</operator>
      <value/>
      <orderBy/>
      <sortOrder/>
      <searchStringValues/>
    </fieldCondition>
  </queryCondition>
  <queryType></queryType>
  <advancedString></advancedString>
  <status></status>
</smartGroupInfo>
</api:upsertSmartGroup>
</soapenv:Body>
</soapenv:Envelope>
```

Response example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>3.1.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:upsertSmartGroupResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>The Smart Group 'smartgroup' is created.</responseDescription>
        <message>contacts found with this query: 2593</message>
      </response>
    </ns2:upsertSmartGroupResponse>
  </soap:Body>
</soap:Envelope>
```

5.6. deleteSmartgroups

Deletes a smartgroup from the contact database.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database
contactGroup	ContactGroup	Properties of the new contact group

Returns:

Api response object

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteSmartGroup>
      <authInfo>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<client>Tripolis Support</client>
<username>support@tripolis.com</username>
<password>*****</password>
</authInfo>
<databaseName>default</databaseName>
<smartGroupName>smartgroup</smartGroupName>
</api:deleteSmartGroup>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>3.1.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:deleteSmartGroupResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>smartgroup 'smartgroup' successfully deleted</message>
      </response>
    </ns2:deleteSmartGroupResponse>
  </soap:Body>
</soap:Envelope>
```

5.7. getSmartgroups

Returns the smartgroups for a given contact database

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database

Returns:

smartgroupInfo object(s)

Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getSmartGroups>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <databaseName>default</databaseName>
    </api:getSmartGroups>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
```


Tripolis Solutions

Tripolis Dialogue Webservices



```
<api-version>3.1.9</api-version>
</soap:Header>
<soap:Body>
  <ns2:getSmartGroupsResponse xmlns:ns2="http://api.tripolis.com/">
    <smartGroupInfo>
      <label>tripolis support</label>
      <name>tripolissupport</name>
      <queryCondition>
        <fieldCondition>
          <fieldName>support</fieldName>
          <operator>is set</operator>
          <value/>
          <orderBy>>false</orderBy>
          <searchStringValue>>false</searchStringValue>
        </fieldCondition>
      </queryCondition>
      <queryType/>
      <advancedString/>
      <status>ACTIVE</status>
    </smartGroupInfo>
    <smartGroupInfo>
      <label>email</label>
      <name>email</name>
      <queryCondition>
        <contactGroupCondition>
          <groupName>emailgroup</groupName>
          <inGroup>>true</inGroup>
          <includeSubgroups>>true</includeSubgroups>
        </contactGroupCondition>
      </queryCondition>
      <queryCondition>
        <fieldCondition>
          <fieldName>email</fieldName>
          <operator>is set</operator>
          <value/>
          <orderBy>>false</orderBy>
          <searchStringValue>>false</searchStringValue>
        </fieldCondition>
      </queryCondition>
      <queryType>ALL</queryType>
      <advancedString/>
      <status>ACTIVE</status>
    </smartGroupInfo>
  </ns2:getSmartGroupsResponse>
</soap:Body>
</soap:Envelope>
```

5.8. importContacts

Imports a csv file to the given database or group. If no contactGroupName is provided the csv file will be imported into the given database. The response is an encrypted export Id which can be used to get the importStatus. Only CSV files are supported.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database
contactGroupName	String	Name of the group

Tripolis Solutions

Tripolis Dialogue Webservices



importMode	String	One of the following: <ul style="list-style-type: none"> - ADD_TO_GROUP - ADD_OR_UPDATE - EXCLUDE_FROM_GROUP - ONLY_ALLOW_ADD - SYNCHRONIZE_GROUP - ONLY_ALLOW_UPDATE - ACCEPTEMAIL_ADD_TO_GROUP (for use in combination with AcceptEmail Integration)
reportReceiver	String	The email recipient for the export report
importFile	Attachment	The file attachment (max. 20MB)

- ADD_TO_GROUP
Add or update contacts to a group
 New contacts from the import file will be added to a group and existing contacts from the import file will be updated.
- ONLY_ALLOW_ADD_TO_GROUP
Add new contact to a group
 Contacts in the import file that are not in a group will be added to a group. Existing contacts will not be imported or updates.
- ADD_OR_UPDATE
Add or update contacts to a database
 New contacts from the import file will be added to the database and existing contacts from the import file will be updated.
- EXCLUDE_FROM_GROUP
Exclude contacts from a group
 Contacts from an import file will be removed from a group.
- ONLY_ALLOW_ADD
Add contacts to a database
 new contacts from the import file will be added to the database and existing records will be ignored.
- SYNCHRONIZE_GROUP
Add or update contacts to a group and synchronize
 New contacts from the import file will be added to a group and existing contacts from the import file will be updated. Contacts in a Group, but not in an import file will be deleted from the group.
- ONLY_ALLOW_UPDATE
Update contacts in a database
 Contacts in the database will be updated, contacts who are not in the database, but in the import file will not be imported or updated.

Returns:

String with an encrypted export Id

Tripolis Solutions

Tripolis Dialogue Webservices



Tripolis Solutions

Tripolis Dialogue Webservices



5.9. getImportStatus

Get the status of a running import.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
importId	String	Encrypted id provided by call importContacts

Returns:

- ImportStatus object

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getImportStatus>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <importId>MTgxNzIxODHEPZFyCM2*0g</importId>
    </api:getImportStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>3.1.13</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getImportStatusResponse xmlns:ns2="http://api.tripolis.com/">
      <importStatus>
        <running>false</running>
        <startTime>2008-11-10T11:09:48+01:00</startTime>
        <endTime>2008-11-10T11:09:52+01:00</endTime>
        <importMode>ONLY_ALLOW_ADD</importMode>
        <created>170</created>
        <updated>0</updated>
        <addedToGroup>0</addedToGroup>
        <removedFromGroup>0</removedFromGroup>
        <numberOfErrors>7</numberOfErrors>
        <importedBy>btoon@tripolis.com</importedBy>
        <importErrors>error at line 125 : INVALID_FIELD_VALUE</importErrors>
        <importErrors>error at line 126 : INVALID_FIELD_VALUE</importErrors>
        <importErrors>error at line 127 : INVALID_FIELD_VALUE</importErrors>
        <importErrors>error at line 128 : INVALID_FIELD_VALUE</importErrors>
        <importErrors>error at line 129 : INVALID_FIELD_VALUE</importErrors>
        <importErrors>error at line 130 : INVALID_FIELD_VALUE</importErrors>
        <importErrors>error at line 131 : INVALID_FIELD_VALUE</importErrors>
      </importStatus>
    </ns2:getImportStatusResponse>
  </soap:Body>
</soap:Envelope>
```

5.10. createDatabase

Create a new database with fields.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseLabel	String	Label for the new database (as shown in application)
databaseName	String	Name of the new database. If no name is given one will be created from the label.
databaseFields	List	A list of DatabaseField objects

Returns:

A success string

5.11. createDatabaseField

Create new field in an existing database.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the new database. If no name is given one will be created from the label.
databaseField	DatabaseField	A DatabaseField object

Returns:

Name of the field

5.12. deleteDatabaseField

Create new field in an existing database.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the database.
databaseFieldName	String	Name of the DatabaseField object that is to be deleted

Returns:

Api response object

Tripolis Solutions

Tripolis Dialogue Webservices



5.13. assignContactsToGroup

Assigns one or more contacts to a group within a specified database.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of the new database. If no name is given one will be created from the label.
groupName	String	A DatabaseField object
refName	String	
refIp	String	
contactIds	List	List of encrypted contact id's (Strings)

Returns:

1. when all contacts were successfully assigned an empty list of KeyValuePairs.
2. when one or more contacts were not assigned a list of KeyValuePairs containing the contactId (key) and the reason of failure (value) for each failed contact.
3. An exception in case something went wrong

Exceptions:

1. InvalidAuthInfoException: in case of wrong credentials.
2. InvalidInputException: in case of a faulty parameter.
3. TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:assignContactsToGroup>

      <authInfo>
        <client>clientname</client>
        <username>username</username>
        <password>password</password>
      </authInfo>
      <databaseName>databasename</databaseName>
      <groupName>groupname</groupName>
      <refName>name of the refferer</refName>
      <refIp>127.0.0.1</refIp>
      <!--Zero or more repetitions:-->
      <contactIds>MjIyMjIyMjKyZkVWQ29IoJA</contactIds>
      <contactIds>MTExMTExMTFecVsa2UKY8g</contactIds>
      <contactIds>NjY2NjY2NjYvqwsaXLa4vvA</contactIds>
      <contactIds>Nzc3Nzc3NzfKFiv3fg_BVg</contactIds>
    </api:assignContactsToGroup>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example 1 :

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.4</api-version>
  </soap:Header>
  <soap:Body>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<ns1:assignContactsToGroupResponse xmlns:ns1="http://api.tripolis.com/" />
</soap:Body>
</soap:Envelope>
```

Response Example 2:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.4</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:assignContactsToGroupResponse xmlns:ns1="http://api.tripolis.com/">
      <contactStatus>
        <key>Mzk3MTkzOTed3E6K5NRDgw</key>
        <value>Contact not found in the specified database.</value>
      </contactStatus>
      <contactStatus>
        <key>Mzk3MTkzOTed3E6K5NRDgw</key>
        <value>Contact not found in the specified database.</value>
      </contactStatus>
    </ns1:assignContactsToGroupResponse>
  </soap:Body>
</soap:Envelope>
```

5.14. getMailings

You are advised to use the calls `getMailJobs` and `getSmsJobs` instead of the `getMailings` call as they provide more information and return only e-mail or sms jobs.

Retrieves all the e-mail and sms jobs that are found within the timerange, that belong to one of the databases that the requesting user has access to.
When the `databaseName` is specified only the mailings for that database will be retrieved.
When the `groupName` is specified only the mailings that were sent to that group will be retrieved.
When the `analyticsIntegrationPartner` is specified only mailings that contained links for that `analyticsIntegrationPartner` will be retrieved.

Parameters:

Name	Type	Description
<code>authInfo</code>	AuthInfo	Object used for authentication of user
<code>timeRange</code>	TimeRange	The range in which the found mailings will be scheduled.
<code>databaseName</code>	String	Name of a database (optional)
<code>groupName</code>	String	Name of a contact group (optional)
<code>analyticsIntegrationPartner</code>	String	The name of the partner e.g. SITESTAT, GOOGLE, GENERIC (optional)

Returns:

1. a list of Mailing objects
2. an empty list

Exceptions:

Tripolis Solutions

Tripolis Dialogue Webservice



4. InvalidAuthInfoException: in case of wrong credentials.
5. InvalidInputException: in case of a faulty parameter.
6. TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMailings>
      <authInfo>
        <client>clientname</client>
        <username>username@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <timeRange>
        <startDate>2007-01-01</startDate>
        <endDate>2008-09-01</endDate>
      </timeRange>
      <databaseName></databaseName>
      <groupName></groupName>
      <analyticsIntegrationPartner></analyticsIntegrationPartner>
    </api:getMailings>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.4</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:getMailingsResponse xmlns:ns1="http://api.tripolis.com/">
      <mailing>
        <mailingId>MTAwMTAwMTC5qOZLQLM0IQ</mailingId>
        <mailingName>Test</mailingName >
        <subject>Test Mailing</subject >
        <startTime>2008-03-26T11:23:43+01:00</startTime>
        <endTime>2008-03-26T11:23:43+01:00</endTime>
        <campaignName/>
        <databaseName>mydatabase</databaseName>
        <groupName>campaigngroup</groupName>
        <smartgroupName/>
        <mailsPerHour>500000</mailsPerHour>
        <mailInterval>0</mailInterval>
        <sampleNumber>0</sampleNumber>
        <status>outbound.label.ended</status>
        <error/>
        <isRemoved>>false</isRemoved>
        <createdBy>username@tripolis.com</createdBy>
        <requestedNumberOfSend>3</requestedNumberOfSend>
        <numberOfSend>2</numberOfSend>
        <numberOfSkipped>1</numberOfSkipped>
      </mailing>
    </ns1:getMailingsResponse>
  </soap:Body>
</soap:Envelope>
```


Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

5.15. getMailJobs

Retrieves all the mail jobs that are found within the timerange, that belong to one of the databases that the requesting user has access to.

When the databaseName is specified only the mailings for that database will be retrieved. When the groupName is specified only the mailings that were sent to that group will be retrieved.

When the analyticsIntegrationPartner is specified only mailings that contained links for that analyticsIntegrationPartner will be retrieved.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
timeRange	TimeRange	The range in which the found mailings will be scheduled.
databaseName	String	Name of a database (optional)
groupName	String	Name of a contact group (optional)
analyticsIntegrationPartner	String	The name of the partner e.g. SITESTAT, GOOGLE, GENERIC (optional)

Returns:

3. a list of mail objects
4. an empty list

Exceptions:

7. InvalidAuthInfoException: in case of wrong credentials.
8. InvalidInputException: in case of a faulty parameter.
9. TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMailJobs>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <timeRange>
        <startDate>2009-10-01</startDate>
        <endDate>2009-12-01</endDate>
      </timeRange>
      <databaseName>tripolis</databaseName>
      <groupName></groupName>
      <analyticsIntegrationPartner></analyticsIntegrationPartner>
    </api:getMailJobs>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<soap:Header>
  <api-version>3.1.13</api-version>
</soap:Header>
<soap:Body>
  <ns2:getMailJobsResponse xmlns:ns2="http://api.tripolis.com/">
    <mailJob>
      <jobId>MTcyOTc4NDcMdCDprzC_oRpaAAO2LvZr</jobId>
      <jobName>subscribers - Newsletter oktober 2009 - 2009-10-29 15:26:06</jobName>
      <startTime>2009-10-29T15:26:06+01:00</startTime>
      <endTime>2009-10-29T15:26:06+01:00</endTime>
      <campaignName/>
      <databaseName>tripolisupdate</databaseName>
      <groupName>subscribers</groupName>
      <smartgroupName/>
      <workspaceName>tripolisupdate</workspaceName>
      <status>ENDED</status>
      <error/>
      <isRemoved>>false</isRemoved>
      <createdBy>support@tripolis.com</createdBy>
      <requestedNumberOfSend>813</requestedNumberOfSend>
      <numberOfSend>739</numberOfSend>
      <numberOfSkipped>74</numberOfSkipped>
      <subject>Newsletter Oktober 2009</subject>
      <newsletterTypeName>nieuwsbrief2009</newsletterTypeName>
      <newsletterName>newsletter_oktober_2009_copy</newsletterName>
      <replyTo/>
      <fromAddress>info@tripolis.com</fromAddress>
      <fromName>Tripolis Newsletter</fromName>
      <mailsPerHour>500000</mailsPerHour>
      <mailInterval>7</mailInterval>
      <sampleNumber>0</sampleNumber>
      <isTest>>false</isTest>
      <isTransactional>>false</isTransactional>
    </mailJob>
  </ns2:getMailJobsResponse>
</soap:Body>
</soap:Envelope>
```

5.16. getSmsJobs

Retrieves all the sms jobs that are found within the timerange, that belong to one of the databases that the requesting user has access to. When the databaseName is specified only the mailings for that database will be retrieved. When the groupName is specified only the mailings that were sent to that group will be retrieved.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
timeRange	TimeRange	The range in which the found mailings will be scheduled.
databaseName	String	Name of a database (optional)
groupName	String	Name of a contact group (optional)

Tripolis Solutions

Tripolis Dialogue Webservices



Returns:

5. a list of Mailing objects
6. an empty list

Exceptions:

10. InvalidAuthInfoException: in case of wrong credentials.
11. InvalidInputException: in case of a faulty parameter.
12. TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getSmsJobs>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <databaseName></databaseName>
      <groupName></groupName>
      <timeRange>
        <startDate>2009-10-01</startDate>
        <endDate>2009-12-01</endDate>
      </timeRange>
    </api:getSmsJobs>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>3.1.13</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getSmsJobsResponse xmlns:ns2="http://api.tripolis.com/">
      <smsJob>
        <jobId>MTkwMzQxODGMdCDprzC_oRpaAAO2LvZr</jobId>
        <jobName>2009 - doen2009 - 2009-11-31 23:59:00</jobName>
        <startTime>2009-11-31T23:59:00+01:00</startTime>
        <endTime>2009-11-31T23:59:00+01:00</endTime>
        <campaignName>2009</campaignName>
        <databaseName>doen2009</databaseName>
        <groupName>campaign_5412_2</groupName>
        <smartgroupName/>
        <workspaceName>doen2009</workspaceName>
        <status>ENDED</status>
        <error/>
        <isRemoved>>false</isRemoved>
        <createdBy/>
        <requestedNumberOfSend>40</requestedNumberOfSend>
        <numberOfSend>40</numberOfSend>
        <numberOfSkipped>0</numberOfSkipped>
        <smsMessageType>oen2009</smsMessageType>
        <smsMessageName>oen2009</smsMessageName>
        <originator>0615295155</originator>
        <description/>
      </smsJob>
    </ns2:getSmsJobsResponse>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
</soap:Body>  
</soap:Envelope>
```

5.17. getMailing

Retrieves a specific Mailing object. The object can be found by mailingId.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	Encrypted id of the mailing

Returns:

- a Mailing object

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:api="http://api.tripolis.com/">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <api:getMailing>  
      <authInfo>  
        <client>clientname</client>  
        <username>username@tripolis.com</username>  
        <password>*****</password>  
      </authInfo>  
      <mailingId>MTA2MTA2MTCIiksgStyMeQ</mailingId>  
    </api:getMailing>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Header>  
    <api-version>1.5.5</api-version>  
  </soap:Header>  
  <soap:Body>  
    <ns1:getMailingResponse xmlns:ns1="http://api.tripolis.com/">  
      <mailing>  
        <mailingId>MTA2MTA2MTCIiksgStyMeQ</mailingId>  
        <mailingName>Confirmation Mail - 2008-07-29 11:36:45</mailingName>  
        <subject>Confirmation Mail</subject>  
        <startTime>2008-07-29T11:36:45+02:00</startTime>  
        <endTime>2008-07-29T11:36:45+02:00</endTime>  
        <campaignName/>  
        <databaseName>mydatabase</databaseName>  
        <groupName>testgroup</groupName>  
        <smartgroupName/>  
        <mailsPerHour>500000</mailsPerHour>  
        <mailInterval>0</mailInterval>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<sampleNumber>0</sampleNumber>
<status>ENDED</status>
<error/>
<isRemoved>>false</isRemoved>
<createdBy>username@tripolis.com</createdBy>
<requestedNumberOfSend>2</requestedNumberOfSend>
<numberOfSend>2</numberOfSend>
<numberOfSkipped>0</numberOfSkipped>
</mailing>
</ns1:getMailingResponse>
</soap:Body>
</soap:Envelope>
```

5.18. getLastMailingSnapshot

Retrieves the last sent MailingSnapshot object. The object can be found by contact group, by type definition or by the combination of both. The MailingSnapshot object is an extension of the Mailing object in the sense that it contains all the fields of the Mailing object plus the html and text content of the mail snapshot.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	String	Name of a database
workspaceName	String	Name of the workspace attached to the database
groupName	String	Name of a contact group (optional)
sampleContactId	String	The encrypted string value of the id of the contact that will be used for showing contact data in the parsed mail snapshot (optional)
mailingType	String	EMAIL or NEWSLETTER
typeDefinitionName	String	Name of the type definition of either the email or the newsletter (optional)

NOTE: although the groupName and typeDefinitionName are optional, at least one of them must be filled in.

Returns:

- a MailingSnapshot object

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getLastMailingSnapshot>
      <authInfo>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<client>clientname</client>
<username>username</username>
<password>*****</password>
</authInfo>
<databaseName>databasename</databaseName>
<workspaceName>workspacename</workspaceName>
<groupName>groupname</groupName>
<sampleContactId>MTc1MTE3NTHW_XmfCILY8A</sampleContactId>
<mailingType>EMAIL</mailingType>
<typeDefinitionName>definitionname</typeDefinitionName>
</api:getLastMailingSnapshot>
</soapenv:Body>
</soap:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.4</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:getLastMailingSnapshotResponse xmlns:ns1="http://api.tripolis.com/">
      <mailingSnapshot>
        <mailingId>Nzg1NTQwNzjzIZ_xvJZt*g</mailingId>
        <mailingName>Uitleg links - 2008-05-26 10:43:00</mailingName>
        <subject>Uitleg links</subject>
        <startTime>2008-05-26T10:43:00+02:00</startTime>
        <endTime>2008-05-26T10:43:00+02:00</endTime>
        <campaignName/>
        <databaseName>testdatabase</databaseName>
        <groupName>mygroup</groupName>
        <smartgroupName/>
        <mailsPerHour>500000</mailsPerHour>
        <mailInterval>0</mailInterval>
        <sampleNumber>0</sampleNumber>
        <status>outbound.label.ended</status>
        <error/>
        <isRemoved>>false</isRemoved>
        <createdBy>me@tripolis.com</createdBy>
        <requestedNumberOfSend>1</requestedNumberOfSend>
        <numberOfSend>1</numberOfSend>
        <numberOfSkipped>0</numberOfSkipped>
        <html><![CDATA[<html xmlns="http://www.w3.org/1999/xhtml">
          <head>
            <title>The html version</title>
          </head>
          <body><p>&nbsp;</p>
            <p><a href="http://www.nu.nl" target="_blank">Nu.nl</a> </p>
            <p><a href="http://www.google.nl" target="_blank">Link2</a></p>
            <p><a href="http://www.tripolis.com" target="_blank">Link3&nbsp;</a></p>
          </body></html>]]>
        </html>
        <text>
          This is the text version of this newsletter
        </text>
      </mailingSnapshot>
    </ns1:getLastMailingSnapshotResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

5.19. getMailingSnapshot

Retrieves any MailingSnapshot object. The object can be found by mailingId. The MailingSnapshot object is an extension of the Mailing object in the sense that it contains all the fields of the Mailing object plus the html and text content of the mail snapshot.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	Encrypted id of the mailing
sampleContactId	String	The encrypted string value of the id of the contact that will be used for showing contact data in the parsed mail snapshot (optional)

Returns:

- a MailingSnapshot object

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMailingSnapshot>
      <authInfo>
        <client>clientname</client>
        <username>username</username>
        <password>*****</password>
      </authInfo>
      <mailingId>MTAwMTAwMTC5qOZLQLM0IQ</mailingId>
      <sampleContactId>NTU1NTU1NTW1DI5txyfMMA</sampleContactId>
    </api:getMailingSnapshot>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:getMailingSnapshotResponse xmlns:ns1="http://api.tripolis.com/">
      <mailingSnapshot>
        <mailingId>MTAwMTAwMTC5qOZLQLM0IQ</mailingId>
        <mailingName>121 - 2008-07-29 10:04:50</mailingName>
        <subject>121</subject>
        <startTime>2008-07-29T10:04:50+02:00</startTime>
        <endTime>2008-07-29T10:04:50+02:00</endTime>
        <campaignName/>
        <databaseName>mydatabase</databaseName>
      </mailingSnapshot>
    </ns1:getMailingSnapshotResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<groupName>testgroup</groupName>
<smartgroupName/>
<mailsPerHour>500000</mailsPerHour>
<mailInterval>0</mailInterval>
<sampleNumber>0</sampleNumber>
<status>ENDED</status>
<error/>
<isRemoved>>false</isRemoved>
<createdBy>username@tripolis.com</createdBy>
<requestedNumberOfSend>2</requestedNumberOfSend>
<numberOfSend>2</numberOfSend>
<numberOfSkipped>0</numberOfSkipped>
<html><![CDATA[<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>The html version</title>
  </head>
  <body><p>&nbsp;</p>
    <p><a href="http://www.nu.nl" target="_blank">Nu.nl</a> </p>
    <p><a href="http://www.google.nl" target="_blank">Link2</a></p>
    <p><a href="http://www.tripolis.com" target="_blank">Link3&nbsp;</a></p>
  </body></html>]]>
</html>
<text>
  This is the text version of this newsletter
</text>
</mailingSnapshot>
</ns1:getMailingSnapshotResponse>
</soap:Body>
</soap:Envelope>
```

5.20. tellAFriend

Allows recipients to send a tell-a-friend mail snapshot to friends. This email will be based upon the email the recipient received. The content of the tell-a-friend mail snapshot can be personalized with additional contact information per 'friend' sent through.

However, this is not required. Also if some profile fields are not available, these fields will be filled with profile data of the original contact.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
contactId	String	The encrypted string value of the id of the contact that will be used for showing contact data in the parsed tell-a-friend mail snapshot
mailingId	string	Encrypted id of the mailing
subject	string	Subject of a tell-a-friend mail snapshot (optional and can be personalized)
message	string	Message of a tell-a-friend mail snapshot (optional and can not be personalized)
tellAFriends	tellAFriends	Complex representing the receiver of a tell-a-friend mail snapshot

Tripolis Solutions

Tripolis Dialogue Webservices



Returns:

- response (message shows the count of recipients)
- a Fault object (in case something went wrong)

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:tellAFriend>
      <authInfo>
        <client>clientname</client>
        <username>username</username>
        <password>*****</password>
      </authInfo>
      <contactId>MTAwMTAwMTC5qOZLQLM0IQ</contactId>
      <mailingId>MTAwMTAwMTC5qOZLQLM0IQ</mailingId>
      <subject>Mail verstuurd door ${sender.name!} naar ${contact.name!}</subject>
    </subject></subject>
    <message>Misschien is deze mail ook iets voor jou?</message>
    <tellAFriends>
      <email>info@tripolis.com</email>
      <profileFields>
        <key>name</key>
        <value>Pino</value>
      </profileFields>
    </tellAFriends>
    <tellAFriends>
      <email>support@tripolis.com </email>
      <profileFields>
        <key>name</key>
        <value>Tinuz</value>
      </profileFields>
    </tellAFriends>
  </api:tellAFriend>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example (correct call):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.5</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:tellAFriendResponse xmlns:ns1="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>2</message>
      </response>
    </ns1:tellAFriendResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



5.21. previewDirectEmail

Retrieves a direct email by direct email ID or workspace name and direct email name. The returned html/text content can be personalized if a contactId is also supplied.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
directEmailName	String	The name of the direct email
directEmailId	String	Encrypted id of the direct email
contactId	String	Encrypted id of the contact

Returns:

- a DirectEmail object

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:previewDirectEmail>
      <authInfo>
        <client>clientname</client>
        <username>username</username>
        <password>*****</password>
      </authInfo>
      <workspaceName>workspaceName</workspaceName>
      <directEmailName>directEmailName</directEmailName>
      <directEmailId></directEmailId>
      <contactId></contactId>
    </api:previewDirectEmail>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:previewDirectEmailResponse xmlns:ns2="http://api.tripolis.com/">
      <directEmail>
        <directEmailId>MTExMTExMTFecVSa2UKY8g</directEmailId>
        <label>directEmailLabel</label>
        <subject>Direct Email Subject</subject>
        <fromName/>
        <fromAddress/>
        <replyTo/>
      </directEmail>
    </ns2:previewDirectEmailResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

```
<html><![CDATA[[#ftl]<html><title>Confirmation Mail</title><body>[#if !onlineViewer]<p>
Click here to [view_online=view online]</p>[#if]<br/>Dit is een confirmation mail
</body></html>]]></html>
<text>Dit is een confirmation (text version)</text>
</directEmail>
</ns2:previewDirectEmailResponse>
</soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



5.22. scheduleMailing

Schedules a direct email or newsletter at the given time. Also capable of sending test mailings with samples from the contact group/smart group.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	string	The name of the database
contactGroupName	String	The name of the contact group
testContactGroupName	String	The name of the test contact group
smartGroupName	String	The name of the smart group
workspaceName	String	The name of the workspace
directEmailName	String	The name of the direct email
newsletterName	String	The name of the newsletter
testMailing	boolean	True if test mailing
sampleRate	int	Number of sample contacts from given contact group or smart group. If sampleRate is 0 then a test mailing will be sent to the testContactGroup without sample contacts. Recipients will receive text and html email.
mailsPerHour	int	Number of mails that will be sent per hour Values: 10000,25000,50000,100000,250000,500000
embargoDate	Calendar	The date when the mailing must be sent (cannot be in the past)

Returns:

- an encrypted jobId

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:scheduleMailing>
      <authInfo>
        <client>clientname</client>
        <username>username</username>
        <password>*****</password>
      </authInfo>
      <databaseName>databasename</databaseName>
      <contactGroupName>contactgroupname</contactGroupName>
    </api:scheduleMailing>
  </soapenv:Body>
</soapenv:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<testContactGroupName>testgroupname</testContactGroupName>
<smartGroupName></smartGroupName>
<workspaceName>myworkspace</workspaceName>
<directEmailName>directemailname</directEmailName>
<newsletterName></newsletterName>
<testMailing>false</testMailing>
<sampleRate>0</sampleRate>
<mailsPerHour>10000</mailsPerHour>
<embargoDate>2008-08-06T13:20:50</embargoDate>
</api:scheduleMailing>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns1:scheduleMailingResponse xmlns:ns1="http://api.tripolis.com/">
      <jobId>MTI2MTI2MTLliluPi1gYKvg</jobId>
    </ns1:scheduleMailingResponse>
  </soap:Body>
</soap:Envelope>
```

Response Example (incorrect call):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.5</api-version>
  </soap:Header>
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>invalid mailingId</faultstring>
      <detail>
        <ns2:InvalidInputException xmlns:ns2="http://api.tripolis.com/">
          <msg>invalid mailingId</msg>
        </ns2:InvalidInputException>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

5.23. scheduleSms

Schedules a sms job at the given time. Also capable of sending test sms job to a test group.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	string	The name of the contact database
contactGroupName	String	The name of the contact group
testContactGroupName	String	The name of the test group

Tripolis Solutions

Tripolis Dialogue Webservices



smartGroupName	String	The name of the smartgroup
workspaceName	String	The name of the workspace
smsMessageName	String	The name of the sms message
smsTypeName	String	The name of the sms message type
testMailing	boolean	True if test sms job
embargoDate	Calendar	The date when the mailing must be sent (cannot be in the past). If omitted the sms will be send directly

Returns:

- an encrypted jobId

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:scheduleSms>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <databaseName>operations</databaseName>
      <contactGroupName>smsgroup</contactGroupName>
      <testContactGroupName>test</testContactGroupName>
      <smartGroupName/>
      <workspaceName>myworkspace</workspaceName>
      <smsMessageName>mysmsmessage</smsMessageName>
      <smsTypeName>smsmessagetype</smsTypeName>
      <testMailing>true</testMailing>
      <embargoDate/>
    </api:scheduleSms>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:scheduleSmsResponse xmlns:ns2="http://api.tripolis.com/">
      <jobId>MTM5NzU2OTeMdCDprzC_oRpaAAO2LvZr</jobId>
    </ns2:scheduleSmsResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



5.24. updateMailingStatus

Updates the status of a mailjob (pause, resume, remove)

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	Encrypted id for the mail job
status	String	Status to set for the mail job (PAUSE, RESUME, REMOVE)

Returns:

- responseCode 100

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:updateMailingStatus>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <mailingId>MTM5NzU0NziMdCDprzC_oRpaAA02LvZr</mailingId>
      <status>REMOVE</status>
    </api:updateMailingStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:updateMailingStatusResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>status of mailing is now REMOVE</message>
      </response>
    </ns2:updateMailingStatusResponse>
  </soap:Body>
</soap:Envelope>
```

5.25. getDeliveredByMailing

Tripolis Solutions

Tripolis Dialogue Webservices



Receive the contacts to which the e-mail is delivered. Optionally one or more contact profile fields can be returned.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	The encrypted mailingId
databaseFieldNames	String	The name of the databaseField

Returns:

- a list of contacts with the requested databaseFieldNames

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getDeliveredByMailing>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <mailingId>MTM5NzUyNjMmdCDprzC_oRpaAAO2LvZr</mailingId>
      <databaseFieldNames>email</databaseFieldNames>
      <databaseFieldNames>boolean</databaseFieldNames>
    </api:getDeliveredByMailing>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getDeliveredByMailingResponse xmlns:ns2="http://api.tripolis.com/">
      <contactInfo>
        <contactId>Mjg1NjAwNjGMdCDprzC_oZ8rP*czsOVN</contactId>
        <keyValuePair>
          <key>email</key>
          <value>support@tripolis.com</value>
        </keyValuePair>
        <keyValuePair>
          <key>boolean</key>
          <value>>false</value>
        </keyValuePair>
      </contactInfo>
    </ns2:getDeliveredByMailingResponse>
  </soap:Body>
</soap:Envelope>
```


5.26. getDeliveredBySmsJob

Receive the contacts to which the SMS is delivered. Optionally one or more contact profile fields can be returned.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
jobId	string	The encrypted SMS jobId
databaseFieldNames	String	The name of the databaseField

Returns:

- a list of contacts with the requested databaseFieldNames

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

5.27. getNotSendByMailing

Get all not sent contacts by mailingID with reason why they were not mailed. Deprecated; you are advised to use the improved of this call: getNotSentByMailing.

5.28. getNotSentByMailing

Receive the skipped records with reason for a specific mailjob. NOTE: use this method in favour of the (deprecated) getNotSendByMailing method. This method's response contains more specific information concerning the reason of the bounce.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	The encrypted mailingId
databaseFieldNames	String	The name of the databaseField

Returns:

- a list of contacts with the requested databaseFieldNames and the reason why they were not sent.

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:api="http://api.tripolis.com/">
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<soapenv:Header/>
<soapenv:Body>
  <api:getNotSentByMailing>
    <authInfo>
      <client> Tripolis Academy</client>
      <username>support@tripolis.com</username>
      <password>*****</password>
    </authInfo>
    <mailingId>NTQ1OTM4OTVR*aGEHx0fgQ</mailingId>
    <databaseFieldNames>email</databaseFieldNames>
  </api:getNotSentByMailing>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.8</api-version>
    <call-duration>0.0.048</call-duration>
  </soap:Header>
  <soap:Body>
    <ns2:getNotSentByMailingResponse xmlns:ns2="http://api.tripolis.com/">
      <notSentContacts>
        <jobId>NTQ1OTM4OTVR*aGEHx0fgQ</jobId>
        <notSentContacts>
          <contactInfo>
            <contactId>MjY1MTQ1MjaMdCDprzC_oWD8aHrKaivw</contactId>
            <keyValuePairs>
              <key>email</key>
              <value>user1@tripolis.com</value>
            </keyValuePairs>
          </contactInfo>
          <notSentReason>
            <toAddress>user1@tripolis.com</toAddress>
            <type>Blacklisted contact</type>
            <description>blacklistreason.bounce.hard</description>
          </notSentReason>
        </notSentContacts>
        <notSentContacts>
          <contactInfo>
            <contactId>MjY1MTQ1MjaMdCDprzC_oYEHuQaIHmKS</contactId>
            <keyValuePairs>
              <key>email</key>
              <value>user3@treepolis.com</value>
            </keyValuePairs>
          </contactInfo>
          <notSentReason>
            <toAddress>user3@treepolis.com</toAddress>
            <type>Blacklisted contact</type>
            <description>blacklistreason.bounce.hard</description>
          </notSentReason>
        </notSentContacts>
      </notSentContacts>
    </ns2:getNotSentByMailingResponse>
  </soap:Body>
</soap:Envelope>
```

5.29. getNotSentBySmsJob

Receive the skipped records with reason for a specific SMS job. This method's response contains more specific information concerning the reason of the bounce.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
jobId	string	The encrypted SMS jobId
databaseFieldNames	String	The name of the databaseField

Returns:

- a list of contacts with the requested databaseFieldNames and the reason why they were not sent.

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

5.30. getClicksByMailing

Get all clicks by mailingID for this mailing. Supplies info about the contact, time they clicked and which url they clicked on.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	The encrypted mailingId
TimeRange	TimeRange	The time bracket for the clicks

Returns:

- a list of clickedUrl

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getClicksByMailing>
      <authInfo>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<client>clientname</client>
  <username>username </username>
  <password>*****</password>
</authInfo>
<mailingId>MTI2MTI2MTLiluPi1gYKvg</mailingId>
<timeRange>
  <startDate>2008-01-01T12:00:00</startDate>
  <endDate>2008-08-20T12:00:00</endDate>
</timeRange>
</api:getClicksByMailing>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getClicksByMailingResponse xmlns:ns2="http://api.tripolis.com/">
      <clickedUrl>
        <browser>firefox</browser>
        <contactId>MTEwMzExMDNIE773fbOq3g</contactId>
        <ipAddress>127.0.0.1</ipAddress>
        <jobId>MTI2MTI2MTLiluPi1gYKvg</jobId>
        <OS>win</OS>
        <timeStamp>2008-08-19T10:30:07+02:00</timeStamp>
        <url>http://www.javaworld.com/javaworld/jw-07-2006/jw-0717-ruby.html</url>
      </clickedUrl>
      <clickedUrl>
        <browser>firefox</browser>
        <contactId>MTEwMzExMDNIE773fbOq3g</contactId>
        <ipAddress>127.0.0.1</ipAddress>
        <jobId>MTI2MTI2MTLiluPi1gYKvg</jobId>
        <OS>win</OS>
        <timeStamp>2008-08-19T10:29:57+02:00</timeStamp>
        <url>https://javapaste.dev.java.net/</url>
      </clickedUrl>
    </ns2:getClicksByMailingResponse>
  </soap:Body>
```

Response Example (incorrect call):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>invalid mailingId</faultstring>
      <detail>
        <ns2:InvalidInputException xmlns:ns2="http://api.tripolis.com/">
          <msg>invalid mailingId</msg>
        </ns2:InvalidInputException>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



5.31. getOpensByMailing

Get all opens by mailingID for this mailing.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	The encrypted mailingId
timeRange	TimeRange	The time bracket for the opens

Returns:

- a list of openInfo

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getOpensByMailing>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <mailingId>MTI2MTI2MTLiluPi1gYKvg</mailingId>
      <timeRange>
        <startDate>2008-01-01T12:00:00</startDate>
        <endDate>2008-08-20T12:00:00</endDate>
      </timeRange>
    </api:getOpensByMailing>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getOpensByMailingResponse xmlns:ns2="http://api.tripolis.com/">
      <openInfo>
        <contactId>MTEwMzExMDNIE773fbOq3g</contactId>
        <jobId>MTI2MTI2MTLiluPi1gYKvg</jobId>
        <ip>127.0.0.1</ip>
        <browser>firefox</browser>
        <timeStamp>2008-08-19T10:29:57+02:00</timeStamp>
      </openInfo>
    </ns2:getOpensByMailingResponse>
  </soap:Body>
```

```
</soap:Envelope>
```

Response Example (incorrect call):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>The startdate must be before the enddate in the timeRange parameter.</faultstring>
      <detail>
        <ns2:InvalidInputException xmlns:ns2="http://api.tripolis.com/">
          <msg>The startdate must be before the enddate in the timeRange parameter.</msg>
        </ns2:InvalidInputException>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

5.32. getCommunicationHistoryByContact

Fetches the communication history for a specific contact.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
contactId	String	The encrypted contactId
workspaceName	String	The name of the workspace
jobId	String	The encrypted jobId
timeRange	TimeRange	The time bracket for the opens
showFullDetails	Boolean	Show detailed response data
includeTransactionalJobs	Boolean	Include transactional response data
includeTestJobs	Boolean	Include testjob response data

Returns:

- a list of mailings that have been sent to the contact
- optionally details about response data can be returned.

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

5.33. exportDeliveredByMailing

Starts an asynchronous export process that will generate an export file (csv) that contains the contacts to which the e-mail has been delivered. The response returns the process id that can be used in combination with. Optionally one or more contact profile fields can be returned.

Tripolis Solutions

Tripolis Dialogue Webservices



authInfo	AuthInfo	Object used for authentication of user
exportId	string	The export process id
compressed	boolean	Compresses the export file (gzip)

Returns:

- the (compressed) export file or the export process status.

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getExportFile>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <exportId>MjMzMDgyMzPo5g1ywecFfA</exportId>
      <compressed>true</compressed>
    </api:getExportFile>
  </soapenv:Body>
</soapenv:Envelope>
```

Reponse Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>3.1.13</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getExportFileResponse xmlns:ns2="http://api.tripolis.com/">
      <exportResponse>
        <apiResponse>
          <responseCode>100</responseCode>
          <responseDescription>success</responseDescription>
          <message>The export has finished</message>
        </apiResponse>
        <exportFile>
          <fileName>exportfile.csv</fileName>
        </exportFile>
      </exportResponse>
    </ns2:getExportFileResponse>
  </soap:Body>
</soap:Envelope>
```

```
<data>H4sIAAAAAAAAAAI1dW3PctPj+31+xdR5TJRu3j2pfaEc6diOM6MoGlvH58UFkhgSJEgwBDmj4a/fbnAkDUA07
ErZSarmA3Fp9L0b/8h0O/Bs+JT/4//+IRou1T/+6x+r6eF6XX06rKvV0PzPe/5OFH90Bfyg6MdxLNukwB+yTdcnP77fe
D/em4E3ydDLTtp3F9PK3/ocs9VLXrq9/e33u+zkhcw39YQIv15CE62bAdb9luHHvRJRjNXbAXomWNOYEtItGYrNTjXo
S/szqsFECVjgMsJjdc6UI+s06d/noxK2F/LFop4CMT5zVrTwHrxXx4nyVDnilZIAPMzAhvQosV1CKDnR2WC8jO728f3i1+
X2ljN0ppnVzqvuCtnIQD+nKxWEcleg27ehCmFEpp7+eLw1CibXk/MMWzshES0AnP4aOZeFn+DFxMLkUU/PB4iiHQ5e
JIONtpk+RnwynZHn/rz6xhOW+AsMI/XxzHHieTi74cs3JIgCIHOE/jzmexiD4VuazaBK9RA//Jnd9fLeYvKvuRNh37ggQt
Fij6/IbueVsiJfjW/f1iXKn21awVFe6TIZS7LhS4uCB/I/kIk0PrIK72JtkJ5XieTND7m8/Xa83d9NqKn/5uvnXevsu294/4
GL61gysGWH03v8OXPNn+NbFeuOBSqDHSreFGXqe8CwT3SBcyP2tB2k4kApc7053nQhigFA8TAXnnvGm47JozdnYKt
nIQeRHInifVcvZmZ/BLbYi38uWbXs4qayGEUQLdzRz57j2QcebtmQZR8TVyt+JUo3D0CCbGdvM//FiC4DULOGefuwv
Eo53F1YfODHa38aZuw6DXyoeHZ/eO9PoZk5qEfox59fL5a3FwYIKLE/bXRbG3fbrxez3os2PyQtN7tVmSDN7y/3xk31
G8XZAq3tBf7xZ17/b0/E2DmA+w1ywWrdVYnncYBjItaUHfUsvSdF7f1ZG9IVjB74384v3y12RjGuUhzYZLrDy/Lfp7
9eUOI+VbpODK7CyJPLdbGuvuwXg2dItvaURJNyYw7u7xebk6d6MIluzxZjHxZUawaxW44+z/7L+RrY1wICPwLxkQ96
y3e6h0+8bqMFvFscbrpTYzEsrYJ/Xj8vdicH3tiPsk3EG0u0P71cjCwY8E+YAuxkcck2uhdvAOCj0+1qgYxtZFYuxPMRsLz
```


Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

```
Vn4BHZZoFbWnrjP1IWjKAEaZCMJgZsiRI4BmNVKGtgT/vFje2BIWs0CIXS/6PiG/7BdOrkXYWt9D++HIxHSXFkDw/m9
OffcuFAtIKUYsEdBANrLuA5Z4gUHT5FNnxUYGAsSmvGjnfBa2a11u6KyUbVvSgUmzsEANXfr/Lfbrt++u27/qvjNzf
3l3/s/tOjliCag1XEgJWAhgEb4Evl1zGeqDGqnc/LX0D3G0mBrI4FXNUIJdvaFZ1H+PNq80h+cxZQQR21qoHp3IDgJVe
ddsDQxIhyWPF24kfkuir2wIT2K3raFdvzvoVbPQKh6LNdK+cpA/R5XT2AUtGQUH3IRM8b0EmFSUo9nBg4AAfm8Q0Up
s/knA+8rTmbgP2nyGvbxKRH6ltPN0AUd0DRgkJ38rkXhWNUgZ4FWsYDsH5yn+Zd/g7WiWwaofxJT99Ad745RCZtZR
UrtYb5gsoL973TwGI5rp0x7kCz+o0aY2ColyqeZLPWf7wLkIMKYHXkt3GmSQnf7WW9lwbtuFfygq3KntfTCXn9+/H38t
</data>
  </exportFile>
</exportResponse>
  </ns2:getExportFileResponse>
</soap:Body>
</soap:Envelope>>
```

5.35. assignArticleToNewsletter

Assigns an existing article to an existing newsletter.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The workspace name
newsletterName	string	The newsletter name
sectionName	string	The section name
articleName	string	The article name

Returns:

- a success message

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:assignArticleToNewsletter>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <newsletterName>mimoneewsletter</newsletterName>
      <sectionName>mainsection</sectionName>
      <articleName>mimoneame</articleName>
    </api:assignArticleToNewsletter>
  </soapenv:Body>
</soapenv:Envelope>
```

5.36. deleteContact

Deletes a contact from a database.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	string	
contactId	string	The encrypted contact id

Returns:

- a success message

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteContact>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <databaseName>mydatabase</databaseName>
      <contactId>NzI5MzU5MTiMdCDprzC_oRpaAAO2LvZr</contactId>
    </api:deleteContact>
  </soapenv:Body>
</soapenv:Envelope>
```

5.37. deleteContactGroup

Deletes a ContactGroup from the database. The contacts that might be in the group will remain in the database.

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	string	
groupName	string	

parameters:

authInfo
databaseName
groupName

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

returns: an ApiResponse containing the ResponseCode.SUCCESS and the text "The contactgroup has been deleted"

5.38. getMaxValueOfContactField

Retrieves the highest value from a specified contact field. For now this field must be of type int.

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
databaseName	string	
fieldName	string	

returns: max value (int)

5.39. getArticle

Retrieves an article from the repository.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
articleId	string	The encrypted article id
contactId	string	The encrypted contact id (optional; enables personalization in the article content)

Returns:

- article id, article label, article name, article modified date, article tags, article fields (including full article html code with (optional) contact personalization).

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getArticle>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <articleId>MTExMTExMTFECVSa2UKY8g</articleId>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<contactId>MTAwMTAwMTC5qOZLQLM0IQ</contactId>
</api:getArticle>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getArticleResponse xmlns:ns2="http://api.tripolis.com/">
      <article>
        <articleId>NjcyNTY3MjUYSzXCnwKgFA</articleId>
        <label>article label</label>
        <name>aticlename</name>
        <modifiedDate>2008-07-21T11:21:05+02:00</modifiedDate>
        <tags>article tag1</tags>
        <tags>article tag2</tags>
        <articleFields>
          <key>title</key>
          <value>Article title</value>
        </articleFields>
        <articleFields>
          <key>content</key>
          <value><![CDATA[<p>Article HTML code</p>]]></value>
        </articleFields>
      </article>
    </ns2:getArticleResponse>
  </soap:Body>
</soap:Envelope>
```

5.40. getBouncesByMailing

Retrieves the bounces for one mailing that were registered within a time range.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	The encrypted mailing id
timeRange	TimeRange	The time range within which the bounces must be registered
databaseFieldNames	List<string>	The field names of the database fields that will be included.

Returns:

- a list of [BouncedContact](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

Tripolis Solutions

Tripolis Dialogue Webservice



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getBouncesByMailing>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <mailingId> MTExMTExMTFEcVSa2UKY8g </mailingId>
      <timeRange>
        <startDate>2008-01-19T10:29:57+02:00</startDate>
        <endDate>2008-08-19T10:29:57+02:00</endDate>
      </timeRange>
      <!--Zero or more repetitions:-->
      <databaseFieldNames>email</databaseFieldNames>
    </api:getBouncesByMailing>
  </soapenv:Body>
</soapenv:Envelope>
```

5.41. getBouncedBySmsJob

Retrieves the bounces for one mailing that were registered within a time range.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
jobId	string	The encrypted SMS job id
timeRange	TimeRange	The time range within which the bounces must be registered
databaseFieldNames	List<string>	The field names of the database fields that will be included.

Returns:

- a list of [BouncedContact](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

5.42. getContactDatabases

Retrieves all contact databases from within one client domain.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user

Returns:

Tripolis Solutions

Tripolis Dialogue Webservices



- a list of [ContactDatabaseInfo](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getContactDatabases>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
    </api:getContactDatabases>
  </soapenv:Body>
</soapenv:Envelope>
```

5.43. getDirectEmailTypes

Retrieves all direct email types within a workspace.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The workspace name

Returns:

- a list of [DirectEmailType](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getDirectEmailTypes>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
    </api:getDirectEmailTypes>
  </soapenv:Body>
```

Tripolis Solutions

Tripolis Dialogue Webservices



</soapenv:Envelope>

5.44. getDirectEmails

Retrieves all direct emails within a workspace, optionally include the html en text content in the response.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The workspace name
emailTypeDefinitionName	String	The name of the direct mail type definition for the directmails
includeContent	Boolean	Include HTML and TEXT content (optional)

Returns:

- a list of DirectEmail objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getDirectEmails>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <workspaceName>operations</workspaceName>
      <emailTypeDefinitionName>mydirectmailtype</emailTypeDefinitionName>
      <includeContent>1</includeContent>
    </api:getDirectEmails>
  </soapenv:Body>
</soapenv:Envelope>
```

Request Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getDirectEmailsResponse xmlns:ns2="http://api.tripolis.com/">
      <directEmail>
        <directEmailId>ODM5NjgzOTZn71DWzUEGtg</directEmailId>
        <label>My Direct Mail</label>
        <name>mydirectmail</name>
        <subject>Tripolis Solutions directmail</subject>
        <fromName>Tripolis Solutions</fromName>
      </directEmail>
    </ns2:getDirectEmailsResponse>
  </soap:Body>
</soap:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<fromAddress>info@tripolis.com</fromAddress>
<replyTo/>
<html>My direct mail HTML</html>
<text>My direct mail TEXT</text>
</directEmail>
</ns2:getDirectEmailsResponse>
</soap:Body>
</soap:Envelope>
```

5.45. getNewsletterSections

Retrieves the sections of a newsletter type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace to which the newsletter type belongs
newsletterTypeName	string	The name of the newsletter type

Returns:

- a list of [NewsletterSectionInfo](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getNewsletterSections>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <newsletterTypeName>mynewslettertypename</newsletterTypeName>
    </api:getNewsletterSections>
  </soapenv:Body>
</soapenv:Envelope>
```

5.46. getNewsletterTypes

Retrieves all newsletter types in a workspace.

Parameters:

Name	Type	Description
------	------	-------------

Tripolis Solutions

Tripolis Dialogue Webservices



TRIPOLIS SOLUTIONS

authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace

Returns:

- a list of [NewsletterType](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getNewsletterTypes>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
    </api:getNewsletterTypes>
  </soapenv:Body>
</soapenv:Envelope>
```

5.47. getNewsletters

Retrieves all newsletters within a workspace (without content).

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The workspace name
newsletterTypeDefinitionName	String	The name of the newsletter type definition for the newsletters

Returns:

- a list of Newsletter objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getNewsletters>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<authInfo>
  <client>Tripolis Academy</client>
  <username>support@tripolis.com</username>
  <password>*****</password>
</authInfo>
<workspaceName>default</workspaceName>
<newsletterTypeDefinitionName>default</newsletterTypeDefinitionName>
</api:getNewsletters>
</soapenv:Body>
</soapenv:Envelope>
```

Reponse Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getNewslettersResponse xmlns:ns2="http://api.tripolis.com/">
      <newsletter>
        <newsletterId>MzU2MjM1NjIDHQf5XRf*Og</newsletterId>
        <label>My Newsletter 1</label>
        <name>mynewsletter1</name>
        <subject>My Newsletter subject</subject>
        <fromAddress>support@tripolis.com</fromAddress>
        <fromName>Tripolis Solutions</fromName>
        <newsletterTypeName>default</newsletterTypeName>
        <modifiedDate>2009-03-23T16:09:06+01:00</modifiedDate>
      </newsletter>
      <newsletter>
        <newsletterId>MzI0NzMyNDfoPC*LMg*dmw</newsletterId>
        <label>My second newsletter</label>
        <name>mysecondnewsletter</name>
        <subject>Second edition</subject>
        <fromAddress>support@tripolis.com</fromAddress>
        <fromName>Tripolis Solutions</fromName>
        <newsletterTypeName>default</newsletterTypeName>
        <modifiedDate>2009-03-23T16:07:25+01:00</modifiedDate>
      </newsletter>
    </ns2:getNewslettersResponse>
  </soap:Body>
</soap:Envelope>
```

5.48. getSmsMessageTypes

Receive the sms message types for a given workspace

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace

Returns:

- sms message type(s)

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.

Tripolis Solutions

Tripolis Dialogue Webservices



- `InvalidInputException`: in case of a faulty parameter.
- `TripolisUnknownException`: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getSmsMessageTypes>
      <authInfo>
        <client>Tripolis Academy </client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <workspaceName>operations</workspaceName>
    </api:getSmsMessageTypes>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:getSmsMessageTypesResponse xmlns:ns2="http://api.tripolis.com/">
      <smsMessageType>
        <label>My sms message type</label>
        <name>mysmsmessagetype</name>
        <defaultOriginator>Tripolis</defaultOriginator>
        <toMobilePhoneNumberFieldName>mobile</toMobilePhoneNumberFieldName>
      </smsMessageType>
    </ns2:getSmsMessageTypesResponse>
  </soap:Body>
</soap:Envelope>
```

5.49. getSummaryReport

Retrieves the summary report of a mailing.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
mailingId	string	The encrypted mailing id

Returns:

- a [MailingSummaryReport](#) object

Exceptions:

- `InvalidAuthInfoException`: in case of wrong credentials.
- `InvalidInputException`: in case of a faulty parameter.
- `TripolisUnknownException`: in case of an unexpected server exception.

Tripolis Solutions

Tripolis Dialogue Webservices



Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getSummaryReport>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <mailingId>MTAxMTAxMTBulSSUb4xBcQ</mailingId>
    </api:getSummaryReport>
  </soapenv:Body>
</soapenv:Envelope>
```

5.50. upsertWorkspace

Create or update a workspace.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceId	String	Encrypted workspace Id (for updating)
contactDatabaseId	String	Encrypted contact database Id
label	String	The label of the workspace
name	String	The name of the workspace (optional)
databaseName	String	The name of the database (optional)
publicDomainName	String	Public domain to be used (optional)
bounceDomainName	String	Bounce domain to be used (optional)

Returns:

- a list of [WorkspaceInfo](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertWorkspace>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <workspaceInfo>
        <workspaceId></workspaceId>
        <contactDatabaseId>MTc1NjMxNzXqBUdjnl1oNA</contactDatabaseId>
      </workspaceInfo>
    </api:upsertWorkspace>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<label>My Workspace</label>
<name>myworkspace</name>
<databaseName></databaseName>
<publicDomainName></publicDomainName>
<bounceDomainName></bounceDomainName>
</workspaceInfo>
</api:upsertWorkspace>
</soapenv:Body>
</soapenv:Envelope>
```

5.51. getWorkspaces

Retrieves the workspaces within a domain.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user

Returns:

- a list of [WorkspaceInfo](#) objects

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getWorkspaces>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
    </api:getWorkspaces>
  </soapenv:Body>
</soapenv:Envelope>
```

5.52. deleteWorkspace

Delete a workspace

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace

Returns:

Tripolis Solutions

Tripolis Dialogue Webservices



- responseCode 100

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteWorkspace>
      <authInfo>
        <client>Tripolis Academy</client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
    </api:deleteWorkspace>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:deleteWorkspaceResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>workspace 'My Workspace' successfully deleted</message>
      </response>
    </ns2:deleteWorkspaceResponse>
  </soap:Body>
</soap:Envelope>
```

5.53. previewNewsletter

Retrieves a newsletter including the parsed, and optionally personalized, content. Note that either the name or the id of the newsletter needs to be available to be able to retrieve it.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
newsletterName	string	The name of the newsletter
newsletterId	string	The encrypted id of the newsletter
contactId	string	The encrypted id of a contact (optional; used for personalization)

Returns:

Tripolis Solutions

Tripolis Dialogue Webservices



- a [NewsletterPreview](#) object

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:previewNewsletter>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspacename</workspaceName>
      <newsletterName>mynewslettername</newsletterName>
      <newsletterId>mynewsletterid</newsletterId>
      <contactId>MTAxMTAxMTBulSSUb4xBcQ</contactId>
    </api:previewNewsletter>
  </soapenv:Body>
</soapenv:Envelope>
```

5.54. upsertDirectEmailType

Inserts or updates a direct mail type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
directEmailType	DirectEmailType	Object containing direct mail type info

Returns:

- success message with direct email type name

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Insert Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertDirectEmailType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>***</password>
      </authInfo>
    </api:upsertDirectEmailType>
  </soapenv:Body>
</soapenv:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
</authInfo>
<workspaceName>default</workspaceName>
<directEmailType>
  <label>direct mail type</label>
  <name>directmailtype</name>
  <fromName>Tripolis Support</fromName>
  <fromAddress>support@tripolis.com</fromAddress>
  <replyTo></replyTo>
  <encoding>UTF8</encoding>
  <emailFieldName>email</emailFieldName>
  <htmlUrl></htmlUrl>
  <textUrl></textUrl>
  <editorEnabled>true</editorEnabled>
</directEmailType>
</api:upsertDirectEmailType>
</soapenv:Body>
</soapenv:Envelope>
```

Update Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertDirectEmailType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <workspaceName>default</workspaceName>
      <directEmailType>
        <label>direct mail type</label>
        <name>directmailtype</name>
        <fromName>Tripolis Solutions Support</fromName>
        <fromAddress>support@tripolis.com</fromAddress>
        <replyTo></replyTo>
        <encoding>UTF8</encoding>
        <emailFieldName>email</emailFieldName>
        <htmlUrl></htmlUrl>
        <textUrl></textUrl>
        <editorEnabled>>false</editorEnabled>
      </directEmailType>
    </api:upsertDirectEmailType>
  </soapenv:Body>
</soapenv:Envelope>
```

Update Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:upsertDirectEmailTypeResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>directEmailType directmailtype updated</message>
      </response>
    </ns2:upsertDirectEmailTypeResponse>
  </soap:Body>
</soap:Envelope>
```


Tripolis Solutions

Tripolis Dialogue Webservices



5.55. upsertNewsletterType

Inserts or updates newsletter type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
newsletterType	NewsletterType	Object containing newsletter type info

Returns:

- success message with newsletter type name

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Insert Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertNewsletterType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>***</password>
      </authInfo>
      <workspaceName>default</workspaceName>
      <newsletterType>
        <label>my newsletter type</label>
        <name>mynewslettertype</name>
        <fromName>Tripolis Support</fromName>
        <fromAddress>support@tripolis.com</fromAddress>
        <replyTo></replyTo>
        <encoding>UTF8</encoding>
        <emailFieldName>email</emailFieldName>
        <defaultSubject>My Newsletter</defaultSubject>
      </newsletterType>
    </api:upsertNewsletterType>
  </soapenv:Body>
</soapenv:Envelope>
```

Update Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertNewsletterType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>***</password>
      </authInfo>
      <workspaceName>default</workspaceName>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<newsletterType>
  <label>my newsletter type</label>
  <name>mynewslettertype</name>
  <fromName>Tripolis Solutions Support</fromName>
  <fromAddress>support@tripolis.com</fromAddress>
  <replyTo></replyTo>
  <encoding>UTF8</encoding>
  <emailFieldName>email</emailFieldName>
  <defaultSubject>My Newsletter update</defaultSubject>
</newsletterType>
</api:upsertNewsletterType>
</soapenv:Body>
</soapenv:Envelope>
```

Update Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:upsertNewsletterTypeResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>newsletterType mynewslettertype updated</message>
      </response>
    </ns2:upsertNewsletterTypeResponse>
  </soap:Body>
</soap:Envelope>
```

5.56. upsertSmsMessageType

Inserts or updates sms message type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
smsMessageType	SmsMessageType	Object containing sms message type info

Returns:

- success message with sms message type name

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Insert Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertSmsMessageType>
      <authInfo>
        <client>Tripolis Support</client>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<username>support@tripolis.com</username>
<password>****</password>
</authInfo>
<workspaceName>default</workspaceName>
<smsMessageType>
  <label>sms message type</label>
  <name>smsmessagetype</name>
  <defaultOriginator>Tripolis</defaultOriginator>
  <toMobilePhoneNumberFieldName>mobile</toMobilePhoneNumberFieldName>
</smsMessageType>
</api:upsertSmsMessageType>
</soapenv:Body>
</soapenv:Envelope>
```

Update Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertSmsMessageType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <workspaceName>default</workspaceName>
      <smsMessageType>
        <label>sms message type</label>
        <name>smsmessagetype</name>
        <defaultOriginator>TripolisSMS</defaultOriginator>
        <toMobilePhoneNumberFieldName>mobile</toMobilePhoneNumberFieldName>
      </smsMessageType>
    </api:upsertSmsMessageType>
  </soapenv:Body>
</soapenv:Envelope>
```

Update Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:upsertSmsMessageTypeResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>smsMessageType smsmessagetype updated</message>
      </response>
    </ns2:upsertSmsMessageTypeResponse>
  </soap:Body>
</soap:Envelope>
```

5.57. deleteDirectEmailType

Deletes a direct email type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user

Tripolis Solutions

Tripolis Dialogue Webservices



workspaceName	string	The name of the workspace
typeName	string	Contains the direct email type name

Returns:

- success message with direct mail type name

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteDirectEmailType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <workspaceName>default</workspaceName>
      <typeName>directmailtype</typeName>
    </api:deleteDirectEmailType>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:deleteDirectEmailTypeResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>DirectEmailType 'directmailtype' successfully deleted</message>
      </response>
    </ns2:deleteDirectEmailTypeResponse>
  </soap:Body>
</soap:Envelope>
```

5.58. deleteNewsletterType

Deletes a newsletter type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
typeName	string	Contains the newsletter type name

Returns:

- success message with newsletter type name

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteNewsletterType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <workspaceName>default</workspaceName>
      <typeName>mynewslettertype</typeName>
    </api:deleteNewsletterType>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:deleteNewsletterTypeResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>NewsletterType 'mynewslettertype' successfully deleted</message>
      </response>
    </ns2:deleteNewsletterTypeResponse>
  </soap:Body>
</soap:Envelope>
```

5.59. deleteSmsMessageType

Deletes a sms message type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
typeName	string	Contains the sms message type name

Returns:

- success message with sms message type name

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.

Tripolis Solutions

Tripolis Dialogue Webservices



- `InvalidInputException`: in case of a faulty parameter.
- `TripolisUnknownException`: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteSmsMessageType>
      <authInfo>
        <client>Tripolis Support</client>
        <username>support@tripolis.com</username>
        <password>****</password>
      </authInfo>
      <workspaceName>default</workspaceName>
      <typeName>smsmessagetype</typeName>
    </api:deleteSmsMessageType>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.5.9</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:deleteSmsMessageTypeResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>SmsMessageType 'smsmessagetype' successfully deleted</message>
      </response>
    </ns2:deleteSmsMessageTypeResponse>
  </soap:Body>
</soap:Envelope>
```

5.60. deleteSmsMessage

Deletes a direct email type.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	String	The name of the workspace
smsMessageType	String	The sms message type name
smsMessageId	String	The encrypted sms message id

Returns:

- success message

Exceptions:

- `InvalidAuthInfoException`: in case of wrong credentials.
- `InvalidInputException`: in case of a faulty parameter.
- `TripolisUnknownException`: in case of an unexpected server exception.

Tripolis Solutions

Tripolis Dialogue Webservices



Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:deleteSmsMessage>
      <authInfo>
        <client>Tripolis Academy </client>
        <username>support@tripolis.com</username>
        <password>*****</password>
      </authInfo>
      <workspaceName>operations</workspaceName>
      <smsMessageType>mysmsmessagetype</smsMessageType>
      <smsMessageId>MTY2MTE2NjFLvie0O9pz8Q</smsMessageId>
    </api:deleteSmsMessage>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:deleteSmsMessageResponse xmlns:ns2="http://api.tripolis.com/">
      <response>
        <responseCode>100</responseCode>
        <responseDescription>success</responseDescription>
        <message>message is successfully deleted</message>
      </response>
    </ns2:deleteSmsMessageResponse>
  </soap:Body>
</soap:Envelope>
```

5.61. upsertArticle

Inserts or updates an article into the repository.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
articleInfo	ArticleInfo	Contains the fields of the article

Returns:

- the encrypted article id

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Insert Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<soapenv:Header/>
<soapenv:Body>
  <api:upsertArticle>
    <authInfo>
      <client>clientname</client>
      <username>username </username>
      <password>*****</password>
    </authInfo>
    <workspaceName>myworkspace</workspaceName>
    <articleInfo>
      <articleId/>
      <label>label1</label>
      <name>label1</name>
      <articleTypeDefinitionName>articledef1</articleTypeDefinitionName>
      <modifiedBy/>
      <modifiedDate/>
      <!--Zero or more repetitions:-->
      <tags>test</tags>
      <!--Zero or more repetitions:-->
      <articleFields>
        <key>body</key>
        <value>myBody</value>
      </articleFields>
    </articleInfo>
  </api:upsertArticle>
</soapenv:Body>
</soapenv:Envelope>
```

Update Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertArticle>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <articleInfo>
        <articleId>MTAxMTAxMTBulSSub4xBcQ</articleId>
        <label>label1</label>
        <name>label1</name>
        <articleTypeDefinitionName>articledef1</articleTypeDefinitionName>
        <modifiedBy/>
        <modifiedDate/>
        <!--Zero or more repetitions:-->
        <tags>mimo</tags>
        <!--Zero or more repetitions:-->
        <articleFields>
          <key>body</key>
          <value>myBody</value>
        </articleFields>
      </articleInfo>
    </api:upsertArticle>
  </soapenv:Body>
</soapenv:Envelope>
```

5.62. upsertImage

Tripolis Solutions

Tripolis Dialogue Webservices



Inserts or updates an image into the repository.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
imageInfo	ImageInfo	Contains the image fields and the content

Returns:

- the encrypted image id

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertImage>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <image>
        <imageId></imageId>
        <attachment>
          <fileName>21cmx29.jpg</fileName>
          <data>cid:349512624184</data>
        </attachment>
        <name>mimoat2tachm2ent2</name>
        <label>Mimo 2At2tach2m2ent</label>
        <description>myDe2scription</description>
        <!--Zero or more repetitions:-->
        <tags>myTag</tags>
      </image>
    </api:upsertImage>
  </soapenv:Body>
</soapenv:Envelope>
```

5.63. upsertDirectMail

Inserts or updates a direct email.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace

Tripolis Solutions

Tripolis Dialogue Webservices



directEmailTypeName	string	The name of the direct email type
directEmail	DirectEmail	Contains the fields and content of the direct email

Returns:

- the encrypted direct email id

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Insert Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertDirectMail>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <directEmailTypeName>emailtypedefinition1</directEmailTypeName>
      <directEmail>
        <directEmailId/>
        <label>label1</label>
        <name>label1</name>
        <subject>Dit is een test</subject>
        <description>myDescription</description>
        <fromName>myfromName</fromName>
        <fromAddress>test@tripolis.com</fromAddress>
        <replyTo>reply@tripolis.com</replyTo>
        <html>test html</html>
        <text></text>
      </directEmail>
    </api:upsertDirectMail>
  </soapenv:Body>
</soapenv:Envelope>
```

Update Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertDirectMail>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <directEmailTypeName>emailtypedefinition1</directEmailTypeName>
      <directEmail>
        <directEmailId>MTE4MTE4MTEIibXXaTtj0g</directEmailId>
        <label> Sals2a2</label>
        <name/>
        <subject>Dit is een test jaap </subject>
        <description>myDescription jaap</description>
      </directEmail>
    </api:upsertDirectMail>
  </soapenv:Body>
</soapenv:Envelope>
```

Tripolis Solutions

Tripolis Dialogue Webservice



```
<fromName>myfromName jaap </fromName>
<fromAddress>jaap@tripolis.com</fromAddress>
<replyTo>jaap@tripolis.com</replyTo>
<html> blällvllvll </html>
<text>Rajd Lotos: Prolog na rozgrzewkÄ™, Å>ciganie w sobotÄ™ Piotr Maciejewski (Mitsubishi Lancfdf.
Rajdu Lotos stajÄ...c sÄ™ jednoczeÅ>nie pierwszym liderem w gronie goÅ>ciw ...
</text>
</directEmail>
</api:upsertDirectMail>
</soapenv:Body>
</soapenv:Envelope>
```

5.64. upsertNewsletter

Inserts or updates a newsletter.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
newsletterInfo	NewsletterInfo	Contains the newsletter info

Returns:

- the encrypted newsletter id

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Insert Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertNewsletter>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <newsletterInfo>
        <newsletterId/>
        <label>mylabel</label>
        <name>myname</name>
        <subject>mySubject</subject>
        <fromAddress/>
        <fromName/>
        <newsletterTypeName>mynewsletter</newsletterTypeName>
        <modifiedDate/>
      </newsletterInfo>
    </api:upsertNewsletter>
  </soapenv:Body>
```

</soapenv:Envelope>

Update Request Example :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertNewsletter>
      <authInfo>
        <client>clientname</client>
        <username>username </username>
        <password>*****</password>
      </authInfo>
      <workspaceName>myworkspace</workspaceName>
      <newsletterInfo>
        <newsletterId>MTAzMTAzMTAWLxRH_9jZiQ</newsletterId>
        <label>mylabel</label>
        <name>myname</name>
        <subject>mySubject</subject>
        <fromAddress/>
        <fromName/>
        <newsletterTypeName>mynewsletter</newsletterTypeName>
        <modifiedDate/>
      </newsletterInfo>
    </api:upsertNewsletter>
  </soapenv:Body>
</soapenv:Envelope>
```

5.65. upsertSmsMessage

Inserts or updates a sms message.

Parameters:

Name	Type	Description
authInfo	AuthInfo	Object used for authentication of user
workspaceName	string	The name of the workspace
smsMessageType	string	The name of the direct email type
smsMessage	SmsMessage	Contains the fields and content of the sms message

Returns:

- the encrypted sms message id

Exceptions:

- InvalidAuthInfoException: in case of wrong credentials.
- InvalidInputException: in case of a faulty parameter.
- TripolisUnknownException: in case of an unexpected server exception.

Request Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.tripolis.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:upsertSmsMessage>
```

Tripolis Solutions

Tripolis Dialogue Webservices



```
<authInfo>
  <client>Tripolis Academy</client>
  <username>support@tripolis.com</username>
  <password>*****</password>
</authInfo>
<workspaceName>operations</workspaceName>
<smsMessageTypeNames>operations</smsMessageTypeNames>
<smsMessage>
  <id/>
  <label>operations</label>
  <name>operations</name>
  <originator>Tripolis</originator>
  <message>My sms message</message>
  <alternativeMessage>My alternative sms message</alternativeMessage>
  <description/>
</smsMessage>
</api:upsertSmsMessage>
</soapenv:Body>
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <api-version>1.6</api-version>
  </soap:Header>
  <soap:Body>
    <ns2:upsertSmsMessageResponse xmlns:ns2="http://api.tripolis.com/">
      <smsMessageId>MTY2MjE2NjKzLWwEjquirA</smsMessageId>
    </ns2:upsertSmsMessageResponse>
  </soap:Body>
</soap:Envelope>
```