

Live Data Solutions: Smart Technology for Data in Motion

# Live Data platform REST API

Technical documentation



Version 2015\_1.8



## Table of Contents

Table of Contents .....	2
1 – Introduction.....	9
2 – Resources .....	9
3 - Methods .....	11
3.1 - Methods for Contacts .....	11
3.1.1 Get Contact by Id.....	11
3.1.2 Get contact by email .....	12
3.1.4 Get all contacts.....	12
3.1.5 Get contacts with filters .....	14
3.1.6 Add / edit contact.....	15
3.1.8 Remove contact.....	16
3.1.9 Unsubscribe contact.....	17
3.1.10 Resubscribe contact .....	18
3.1.11 Activate contact .....	19
3.2 – Methods for Unsubscribes .....	19
3.2.1 Get Unsubscribes .....	19
3.3 – Methods for Bounces .....	21
3.3.1 Get Bounces .....	21
3.3.2 Get Bounces by Schedule .....	22
3.4 – Methods for FBLs.....	23
3.4.1 Get FBLs.....	23
3.5 – Methods for Receivers .....	24
3.5.1 Get Receivers.....	24
3.6 – Methods for Opens .....	26
3.6.1 Get Opens.....	26



3.6.2 Get Clicks .....	27
3.7 – Methods for segmentations .....	28
3.7.1 Get Segmentations .....	28
3.7.2 Create segment from list of contacts .....	29
3.8 – Methods for List of Contacts .....	30
3.8.1 Get Lists of Contacts .....	30
3.8.2 Get List of Contacts by Id .....	31
3.8.3 Get contacts from List .....	32
3.8.4 Create new List of Contacts .....	33
3.8.5 Remove a List of Contacts .....	35
3.8.6 Add Contact to list .....	35
3.8.7 Remove Contact from list .....	36
3.9 – Methods for Forms .....	37
3.9.1 Get HTML code from Form .....	37
3.10 – Methods for Transactional messages .....	38
3.10.1 Send transactional message .....	38
3.11 – Methods for Drafts .....	39
3.11.1 Get draft by Id .....	39
3.11.2 Create a draft .....	40
3.11.3 Edit draft .....	41
3.12 – Methods for Schedules .....	42
3.12.1 Schedule single message .....	42
3.12.2 Schedule transactional .....	44
3.12.3 Get schedule info by Id .....	46
3.12.4 Get schedule info by tag .....	47
3.12.5 Get list of schedules .....	47
3.12.6 Get list of schedules with filters .....	49
3.12.7 Send Bat to email .....	50



3.12.8 Send Bat lo List of Contacts .....	51
3.12.9 Stop schedule .....	52
3.12.10 Restart schedule .....	53
3.12.11 Remove schedule .....	55
3.13 - Methods for Sender Names .....	56
3.13.1 Get Sender Names.....	56
3.13.2 Get sender name by id .....	57
3.13.3 Get default sender name .....	58
3.13.4 Set default sender name .....	59
3.13.5 Create new sender name .....	60
3.14 - Methods for Statistics .....	61
3.14.1 Get Statistics Summary for single campaigns .....	61
3.14.2 Get Statistics Summary for triggers .....	62
3.14.3 Get Statistics Summary for transactional messages .....	63
3.14.4 Get Statistics Summary for single campaign (by schedule Id) .....	65
3.14.5 Get Statistics Summary for single campaign (by schedule Tag).....	66
3.14.6 Get Statistics Summary for transactional message (by schedule Id) .....	67
3.14.7 Get Statistics Summary for transactional message (by schedule Tag).....	68
3.14.8 Get Statistics Summary for trigger (by schedule Id).....	68
3.14.9 Get Statistics Summary for trigger (by schedule Tag) .....	69
3.14.10 Get statistics by origin .....	70
3.15 Methods for headers.....	71
3.15.1 Get headers .....	71
3.15.2 Get header by id.....	73
3.15.3 Get default header .....	74
3.15.4 Set default header.....	75
3.16 Methods for footers .....	76
3.16.1 Get footers .....	76



3.16.2 Get footer by id .....	77
3.16.3 Get default footer .....	78
3.16.4 Set default footer .....	79
3.17 Methods for Replies to .....	80
3.17.1 Get Replies to .....	80
3.17.2 Set default reply to .....	81
3.17.3 Get reply to by id .....	82
3.17.4 Get default reply to .....	83
3.18 Methods for Tracking Domains .....	84
3.18.1 Get Tracking Domains .....	84
3.18.2 Get tracking domain by id .....	85
3.18.3 Get default tracking domain .....	86
3.18.4 Set default tracking domain .....	87
3.19 Methods for Domains .....	88
3.19.1 Get Domains .....	88
3.19.2 Get domain by id .....	89
3.19.3 Get default domain .....	90
3.19.4 Set default domain .....	91
3.20 Methods for Split Sendouts .....	92
3.20.1 Create new split sendout .....	92
3.21 – Methods for intelligent delivery modes .....	94
3.21.1 Get Intelligent Delivery mode .....	94
3.22 Methods for Attributes .....	95
3.22.1 Get Attributes .....	95
3.22.2 Get attribute by id .....	96
3.22.3 Get Allowed Attribute's Data types .....	97
3.22.4 Create new Custom Attribute .....	98
3.23 Methods for Scenario Manager .....	99



3.23.1 Add contact to program .....	99
3.23.2 Remove contact from program .....	100
4 – Authentication.....	101
5 – Examples.....	102
5.1 Basic script structure (Get segmentation query string) .....	102
5.2 Getting contact(s).....	103
5.2.1 Get contact by email .....	103
5.2.2 Get contact by id .....	103
5.2.3 Get contacts .....	104
5.2.4 Get Unsubscribes / opens / bounces / clicks.....	105
5.3 Managing contacts (add, edit, unsubscribe, remove).....	106
5.3.1 Add / update contact to database .....	106
5.3.2 Delete contact from database.....	107
5.3.3 Unsubscribe contact.....	107
5.3.4 Resubscribe contact .....	108
5.3.5 Activate contact in status queued.....	109
5.4 Managing Segmentations.....	109
5.4.1 Get Segmentations.....	109
5.4.2 Generate segmentation from list of contacts .....	110
5.5 Managing Lists of contacts.....	110
5.5.1 Get lists of contacts from database .....	110
5.5.2 Get list from list Id .....	111
5.5.3 Get contacts from list .....	111
5.5.4 Create a new list of contacts.....	112
5.5.5 Remove a list of contacts .....	113
5.5.6 Add contact to list .....	113
5.5.7 Remove contact from list .....	114
5.6 Custom forms.....	114



5.6.1 Get HTML code for selected form .....	114
5.7 Sender Names management.....	115
5.7.1 Get sender names from database .....	115
5.8 Sending BATs .....	115
5.8.1 Send BAT to email .....	115
5.8.2 Send BAT to list .....	116
5.9 Get schedule(s) info (Single campaigns, triggers or transactionals) .....	117
5.9.1 Get selected schedule .....	117
5.9.2 Get a list of schedules .....	117
5.10 Schedule one campaign .....	118
5.11 Stop schedule .....	118
5.12 Restart stopped schedule.....	119
5.13 Remove schedule .....	119
5.14 Send a transactional message .....	120
5.15 Get Statistics Summary .....	121
5.15.1 – Get Statistics by tag or id.....	121
5.15.2 – Get Statistics with filters .....	121
5.15.3 – Get Statistics by origin .....	122
5.16 Methods for headers.....	123
5.16.1 Get all headers .....	123
5.16.2 Get header by id.....	123
5.16.3 Get default header .....	124
5.17 Methods for footers .....	124
5.17.1 Get all footers.....	124
5.17.2 Get footer by id .....	125
5.17.3 Get default footer .....	125
5.18 Methods for drafts .....	126
5.18.1 Create draft .....	126



5.19 Methods for “Replies to” .....	128
5.19.1 Get all replies to .....	128
5.20 Methods for Split sendout .....	128
5.20.1 Create split sendout .....	128
5.21 Methods for Scenario Manager .....	129
5.21.1 Add contact to program .....	129
5.21.2 Remove contact to program .....	130





## 1 – Introduction

REST (Representational State Transfer) is a model oriented to resources. Each url represents an object or resource.

A RESTful web API (also called a RESTful web service) is a web API implemented using HTTP and the principles of REST. It is a collection of resources, with four defined aspects:

- the base URI for the web API, such as <http://example.com/resources/>
- The Internet media type of the data supported by the web API. This is often JSON but can be any other valid Internet media type provided that it is a valid hypertext standard.
- The set of operations supported by the web API using HTTP methods (e.g., GET, PUT, POST, or DELETE).
- The API must be hypertext driven.

The new Live Data platform API is a RESTful web service that allow the most common actions that can be performed from the web application like contacts management, schedule campaigns, get results of campaigns... . This service is under constant development, so new features will be added.

## 2 – Resources

Below is a list of resources exposed in the Live Data Platform REST API. You must replace the **www[XXX]** by the server in which your database is hosted, and the **[API\_ID\_XXXX]** by an appropriate value for the database you try to connect.

Resources	URL
<b>Contacts</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/contacts">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/contacts</a>
<b>Unsubscribes</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/unsubscribes">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/unsubscribes</a>
<b>Bounces</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/bounces">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/bounces</a>



<b>FBLs</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/fbIs">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/fbIs</a>
<b>Opens</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/opens">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/opens</a>
<b>Segmentations</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/segments">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/segments</a>
<b>List of Contacts</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/contact_lists">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/contact_lists</a>
<b>Sender Names</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/sender_names">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/sender_names</a>
<b>Transactionals</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/transactionals">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/transactionals</a>
<b>Drafts</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/drafts">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/drafts</a>
<b>Schedules</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/schedules">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/schedules</a>
<b>Forms</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/forms">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/forms</a>
<b>Statistics</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/stats">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/stats</a>
<b>Headers</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/headers">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/headers</a>
<b>Footers</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/footers">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/footers</a>
<b>Replies to</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/replies_to">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/replies_to</a>
<b>Split sendouts</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/split_sendouts">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/split_sendouts</a>
<b>Intelligent delivery</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/intelligent_delivery">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/intelligent_delivery</a>
<b>Attributes</b>	<a href="http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/attributes">http://www[XXX].livedata-emailing.com/sliprestful/bases/[API_ID_XXXX]/attributes</a>



## 3 - Methods

### 3.1 - Methods for Contacts

#### 3.1.1 Get Contact by Id

This action returns all the attributes for the selected contact Id.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>CONTACT_ID</b>	Integer	Contact identifier

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/\[CONTACT\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/[CONTACT_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contact found
- 404 - Contact or database not found
- 400 - Invalid type for contact parameter
- 403 – Access forbidden
- 405 – Invalid HTTP method

Response->status: string with: ok or error

Response->result: array with contact attributes if the contact is found, empty array if not.

Response->message: brief description with the execution results



### 3.1.2 Get contact by email

This action returns all the attributes for the received email.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>CONTACT_EMAIL</b>	Integer	Contact email

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/\[CONTACT\\_EMAIL\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/[CONTACT_EMAIL])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contact found
- 400 - Invalid type for contact parameter
- 403 – Access forbidden
- 404 – Contact not found or database not found
- 405 – Invalid HTTP method

Response->status: string with: ok or error

Response->result: array with contact attributes if the contact is found, empty array if not.

Response->message: brief description with the execution results

### 3.1.4 Get all contacts

This action returns all contacts in the database. The results are paginated, and the maximum number of contacts per page is 1000.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>P</b>	Integer	page

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts?p=2](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts?p=2)

**Response format:** json**Response Object detail:**

Response-&gt;statusCode:

- 200 – Contact list found
- 400 - Invalid type for contact parameter
- 403 – Access forbidden
- 404 – Database not found
- 405 – Invalid HTTP method

Response-&gt;status: string with: ok or error

Response-&gt;result-&gt;data: Array of contacts (id, email and status)

Response-&gt;result-&gt;total: integer with the total number of contacts

Response-&gt;result-&gt;page: number of current page

Response-&gt;result-&gt;totalPages: total number of pages

Response-&gt;result-&gt;links-&gt;urlNextPage: Url for the next page of results

Response-&gt;result-&gt;links-&gt;urlPreviousPage: Url for the previous page of results

Response-&gt;message: brief description with the execution results



### 3.15 Get contacts with filters

This action returns all contacts that match with the filters.

#### Parameters:

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>STANDARD OR CUSTOM CONTACT ATTRIBUTE LIST (HTTP GET parameters)</b>		One or more contact attribute using GET HTTP method. (?field1=value1&field2=value2...)

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts?firstname=julio&title=mr](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts?firstname=julio&title=mr)

**Response format:** json

#### Response Object detail:

Response->statusCode:

- 200 - Contact list found
- 400 - Invalid type for contact parameter
- 403 – Access forbidden
- 404 – Database not found
- 405 – Invalid HTTP method

Response->status: string with: ok or error

Response->result->data: Array of contacts (id, email and status)

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages



Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

Response->message: brief description with the execution results

### 3.1.6 Add / edit contact

Insert a new contact in the database or update an existing one.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>EMAIL</b>	String	Contact email
<b>STANDARD OR CUSTOM CONTACT ATTRIBUTE</b>		One or more contact attribute using PUT HTTP method.
<b>TIMEZONE</b>	String	Valid time zone string, like Europe/Paris (*)
<b>LIST_ID</b>	Integer	Id of existing list of contacts where the contact will be added after be added to the database (**)

(\*) If is set, the application understand that all datetime attributes are in this time zone. You can check a valid time zones list in:

[http://en.wikipedia.org/wiki/List\\_of\\_tz\\_database\\_time\\_zones](http://en.wikipedia.org/wiki/List_of_tz_database_time_zones)

(\*\*) This parameter is not mandatory

**Http method:** PUT

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/contact\\_email@domain.com](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/contact_email@domain.com)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Existing contact updated

201 - New contact added



400 - Invalid type for contact parameter

403 - Access forbidden

404 - Database not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->links->contact\_url: link to the added/edited contact

Response-result-> empty array

### 3.1.8 Remove contact

This action remove a contact if exists.

#### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>CONTACT EMAIL</b>	String	Contact email

**Http method:** DELETE

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/contact\\_email@domain.com](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/contact_email@domain.com)

**Response format:** json

#### **Response Object detail:**

Response->statusCode:

200 - Contact removed

400 - Invalid type for contact parameter

403 - Access forbidden





404 - Database or contact not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response-result-> empty array

### 3.1.9 Unsubscribe contact

Unsubscribe contact action.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>EMAIL</b>	String	Contact email

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/contact\\_email@domain.com/unsubscription](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/contact_email@domain.com/unsubscription)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Contact unsubscribed

400 - Invalid type for contact parameter

403 - Access forbidden

404 - Database or contact not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results



Response-result-> empty array

Response->links->contactUrl: link to the unsubscribed contact

### 3.1.10 Resubscribe contact

This action set to active an existing unsubscribed contact.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>EMAIL</b>	String	Contact email

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/contact\\_email@domain.com/resubscription](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/contact_email@domain.com/resubscription)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contact resubscribed
- 400 - Invalid type for contact parameter
- 403 - Access forbidden
- 404 - Database or contact not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response-result-> empty array

Response->links->contactUrl: link to the resubscribed contact



### 3.1.11 Activate contact

This action set to active an existing contact in status “queued”.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>EMAIL</b>	String	Contact email

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contacts/contact\\_email@domain.com/activation](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contacts/contact_email@domain.com/activation)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contact activated
- 400 - Invalid type for contact parameter
- 403 - Access forbidden
- 404 - Database or contact not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result: empty array

Response->links->contactUrl: link to the activated contact

## **3.2 – Methods for Unsubscribes**

### 3.2.1 Get Unsubscribes

This action returns a list of unsubscribers that match with the criteria. If none criterion is received, return all unsubscribers



Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FILTER LIST (HTTP GET parameters)</b>		Filter list in GET HTTP method format. List of allowed filters: <ul style="list-style-type: none"> <li>- P: page</li> <li>- fromDate</li> <li>- toDate</li> <li>- timezone</li> <li>- dateFormat</li> </ul>

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/unsubscribes?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&timezone=Europe/London&dateFormat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/unsubscribes?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&timezone=Europe/London&dateFormat=yyyy-mm-dd)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contacts found
- 400 - Invalid type for contact parameter
- 403 - Access forbidden
- 404 - Database or contact not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of contacts (id, email and status)

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages



Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### 3.3 – Methods for Bounces

#### 3.3.1 Get Bounces

This action returns a list of bounces that match with the criteria. If none criterion is received, return all bounces

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FILTER LIST(HTTP GET parameters)</b>		Filter list in GET HTTP method format. List of allowed filters: <ul style="list-style-type: none"> <li>- P: page</li> <li>- fromDate</li> <li>- toDate</li> <li>- timezone</li> <li>- dateFormat</li> </ul>

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/bounces?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&timezone=Europe/London/dateFormat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/bounces?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&timezone=Europe/London/dateFormat=yyyy-mm-dd)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Contacts found

400 - Invalid type for contact parameter

403 - Access forbidden



404 - Database or contact not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of contacts (id, email and status)

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### 3.3.2 Get Bounces by Schedule

This action returns a list of emails in status bounced for the selected schedule. Both, schedule id and tag are allowed for filtering.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SCHEDULE_ID OR TAG</b>	String	Schedule identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_ID\]/bounces?p=XX](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_ID]/bounces?p=XX)

**Response format:** json

**Response Object detail:**

Response->statusCode:



200 - Contacts found

400 - Invalid parameter

403 - Access forbidden

404 - Database or schdeule not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of emails

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### ***3.4 – Methods for FBLs***

#### **3.4.1 Get FBLs**

This action returns a list of fbls for the selected schedule.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FILTER LIST (HTTP GET parameters)</b>		Filter list in GET HTTP method format. List of allowed filters: <ul style="list-style-type: none"> <li>- P: page</li> <li>- schedule (scheduleId or tag)</li> </ul>



**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/fbIs?schedule=data\\_xyyzzz](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/fbIs?schedule=data_xyyzzz)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contacts found
- 400 - Invalid parameter
- 403 - Access forbidden
- 404 - Database or contact not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of emails

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

## ***3.5 – Methods for Receivers***

### ***3.5.1 Get Receivers***

This action returns a list of those contacts that received the selected campaign





Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FILTER LIST (HTTP GET parameters)</b>		List of allowed GET filters: - schedule (tag)

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/receivers?schedule=data xxxyyzz](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/receivers?schedule=data xxxyyzz)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contacts found
- 400 - Invalid parameter or parameter not found
- 403 - Access forbidden
- 404 - Database or schedule not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of contacts

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results



### 3.6 – Methods for Opens

#### 3.6.1 Get Opens

This action returns a list of opens that match with the criteria. If none criterion is received, return all opens

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FILTER LIST (HTTP GET parameters)</b>		List of allowed GET filters: <ul style="list-style-type: none"> <li>- P: page</li> <li>- fromDate</li> <li>- toDate</li> <li>- schedule (scheduleId or tag)</li> <li>- dateFormat</li> </ul>

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/opens?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&schedule=123&dateFormat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/opens?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&schedule=123&dateFormat=yyyy-mm-dd)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 - Contacts found
- 400 - Invalid type for contact parameter
- 403 - Access forbidden
- 404 - Database or contact not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error



Response->message: brief description with the execution results

Response->result->data: Array of contacts (id, email and status)

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### 3.6.2 Get Clicks

This action returns a list of clicks that match with the criteria. If none criterion is received, return all clicks

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FILTER LIST (HTTP GET parameters)</b>		Filter list in GET HTTP method format. List of allowed filters: <ul style="list-style-type: none"> <li>- P: page</li> <li>- fromDate</li> <li>- toDate</li> <li>- schedule (scheduleId or tag)</li> <li>- dateFormat</li> </ul>

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/clicks?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&schedule=123&dateFormat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/clicks?fromDate=2012-03-02+10:00:00&toDate=2012-04-30+12:15:00&p=1&schedule=123&dateFormat=yyyy-mm-dd)

**Response format:** json

**Response Object detail:**

Response->statusCode:



200 - Contacts found

400 - Invalid type for contact parameter

403 - Access forbidden

404 - Database or contact not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of contacts (id, email and status)

Response->result->total: integer with the total number of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### ***3.7 – Methods for segmentations***

#### **3.7.1 Get Segmentations**

This action returns a list of segmentations for the selected database.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/segments](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/segments)

**Response format:** json



Response->statusCode:

200 - Segments found

403 - Access forbidden

404 - Database or segmentations not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of segmentations. Each segment has the properties:

- id: Integer
- name: String
- description: String
- user: String
- creation: Datetime (referred to timezone)
- contactsEstimation (relating to the lastCalculation field): Integer
- lastCalculation: Datetime (referred to timezone)
- timezone: String

Response->result->total: integer with the total number of segmentations

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.7.2 Create segment from list of contacts

This action generates a new segmentation from the selected list of contacts

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>LIST_ID</b>	Integer	List identifier

**Http method:** PUT



URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/segments/contact\\_lists/\[LIST\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/segments/contact_lists/[LIST_ID])

**Response format:** json

Response->statusCode:

- 200 – Segmentation has been generated
- 400 - Validation error
- 403 - Access forbidden
- 404 - Database or lists of contacts not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result-> Segmentation Id

### ***3.8 – Methods for List of Contacts***

#### ***3.8.1 Get Lists of Contacts***

This action returns the lists of the contacts for the selected database.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists)

**Response format:** json

Response->statusCode:

- 200 - Lists of contacts found



403 - Access forbidden

404 - Database or lists of contacts not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of lists of contacts. The properties for the each list of contact:

- id: Integer
- name: String
- description: String
- base: String
- batList: Boolean
- creation: Datetime(referred to timezone)
- timezone: String
- links
  - getContacts: url to obtain the contacts in the list

Response->result->total: number of lists of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.8.2 Get List of Contacts by Id

This action returns the selected list of contacts.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>LIST_ID</b>	Integer	List identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists/\[LIST\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists/[LIST_ID])



**Response format:** json

Response->statusCode:

- 200 – List of contacts found
- 403 - Access forbidden
- 404 - Database or lists of contacts not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of lists of contacts. The properties for the list of contact:

- id: Integer
- name: String
- description: String
- base: String
- batList: Boolean
- creation: Datetime (referred to timezone)
- timezone: String
- links
  - getContacts: url to obtain the contacts in the list

Response->result->total: number of lists of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.8.3 Get contacts from List

This action returns the contacts in the selected list

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier





<b>LIST_ID</b>	Integer	List identifier
----------------	---------	-----------------

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists/\[LIST\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists/[LIST_ID])

**Response format:** json

Response->statusCode:

- 200 – Contacts found
- 403 - Access forbidden
- 404 - Database or lists of contacts not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of lists of contacts. The properties for the list of contact:

- id: Integer
- email: String
- status: String

Response->result->total: number of lists of contacts

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### **3.8.4 Create new List of Contacts**

Generate a new list of contacts.



Parameter	Type	Description
<b>DATABASE_API_ID (*)</b>	String	Database identifier
<b>NAME (*)</b>	String	List name. Must be unique
<b>DESCRIPTION (*)</b>	String	Description
<b>BATLIST (*)</b>	Boolean (true / false)	Bat lists are normally used to send tests
<b>EMAILLIST (*)</b>	String	Emails to be added to the list, separated by “,”

(\*) Mandatory fields

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - List has been created

400 - Validation errors

403 - Access forbidden

404 - Database not found

405 - Invalid HTTP method

500 - Internal server error

Response->status: ok or error

Response->message: brief description with the execution results

Response->result: List Id

Response->invalidEmails: array of not added emails due to validation errors



### 3.8.5 Remove a List of Contacts

Remove a list of contacts. Contacts in list are not removed from database.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>LIST_ID</b>	Integer	List identifier

**Http method:** DELETE

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists/\[LIST\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists/[LIST_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – List of contacts has been removed
- 400 – Unexpected error while removing the list
- 403 - Access forbidden
- 404 - Database or list not found
- 405 - Invalid HTTP method
- 500 – Internal server error

Response->status: ok or error

Response->message: brief description with the execution results

Response-result-> empty

### 3.8.6 Add Contact to list

Add a contact to the selected list. Both, the contact and the list must exist in the database.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier



<b>LIST_ID</b>	Integer	List identifier
<b>EMAIL</b>	String	Existing email in database

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists/\[LIST\\_ID\]/contact/\[EMAIL\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists/[LIST_ID]/contact/[EMAIL])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – Contact added to list
- 400 – Unexpected error while adding the contact to the list
- 403 - Access forbidden
- 404 - Contact or list not found
- 405 - Invalid HTTP method
- 500 – Internal server error

Response->status: ok or error

Response->message: brief description with the execution results

Response-result-> empty

### **3.8.7 Remove Contact from list**

Remove one contact from the selected list. Both, the contact and the list must exist in the database.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>LIST_ID</b>	Integer	List identifier
<b>EMAIL</b>	String	Existing email in database



**Http method:** DELETE

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/contact\\_lists/\[LIST\\_ID\]/contact/\[EMAIL\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/contact_lists/[LIST_ID]/contact/[EMAIL])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – Contact removed from list
- 400 – Unexpected error while removing the contact to the list
- 403 - Access forbidden
- 404 - Contact or list not found
- 405 - Invalid HTTP method
- 500 – Internal server error

Response->status: ok or error

Response->message: brief description with the execution results

Response-result-> empty

### ***3.9 – Methods for Forms***

#### ***3.9.1 Get HTML code from Form***

Get HTML code for the selected custom form

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>FORM_ID</b>	Integer	Form identifier

**Http method:** GET



URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/forms/\[FORM\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/forms/[FORM_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Form has been found

400 – Unexpected error

403 - Access forbidden

404 - Database or transactional message not found

405 - Invalid HTTP method

500 – Internal server error

Response->status: ok or errorResponse->message: brief description with the execution results

Response-result-> HTML code

### ***3.10 – Methods for Transactional messages***

#### ***3.10.1 Send transactional message***

Send transactional message

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRANSACTIONAL_ID</b>	Integer	Transactional message identifier
<b>EMAIL (HTTP POST method)</b>	String	By POST

**Http method:** POST

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/transactionals/\[TRANSACTIONAL\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/transactionals/[TRANSACTIONAL_ID])



**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Transactional message has been sent

400 - Invalid type for contact parameter

403 - Access forbidden

404 - Database or transactional message not found

405 - Invalid HTTP method

500 – Internal server error

Response->status: ok or error

Response->message: brief description with the execution results

Response-result-> empty array

### ***3.11 – Methods for Drafts***

#### **3.11.1 Get draft by Id**

This action returns the properties for the selected draft

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_ID</b>	Integer	Draft identifier

**Http method:** GET



**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/drafts/\[DRAFT\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/drafts/[DRAFT_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Draft found

403 – Access forbidden

404 – Draft or database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->result: array with draft data if exists, empty array if not.

Response->message: brief description with the execution results

### **3.11.2 Create a draft**

Generate a new draft

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>NAME (*)</b>	String	Name of the draft
<b>SUBJECT (*)</b>	String	Subject for the draft
<b>SENDER (*)</b>	String	Name of the sender
<b>HTMLCONTENT (*)</b>	String	HTML code for the draft
<b>TEXT CONTENT (*)</b>	String	Text version
<b>KEYWORDS</b>	String	keywords list, separated by ','

(\*) Mandatory fields

**Http method:** PUT





URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/drafts/creation](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/drafts/creation)

**Response format:** json

**Response Object detail:**

Response->statusCode:

201 - Draft created

400 – Validation errors

403 - Access forbidden

404 – Database not found

405 - Invalid HTTP method

Response->status: ok or error

Response->message: brief description with the execution results

Response->result: draft Id

Response-> w3cValidatorResults: W3C validation results

Response-> spamTestResults: Spam test validation results

### 3.11.3 Edit draft

Edit selected draft

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_ID</b>	Integer	Draft identifier
<b>NAME (*)</b>	String	Name of the draft
<b>SUBJECT (*)</b>	String	Subject for the draft
<b>SENDER (*)</b>	String	Name of the sender
<b>HTMLCONTENT (*)</b>	String	HTML code for the draft
<b>TEXT CONTENT (*)</b>	String	Text version



<b>KEYWORDS</b>	String	keywords list, separated by ','
-----------------	--------	---------------------------------

(\*) Mandatory fields

**Http method:** PUT

URL: [http://www\[XXX\].slip-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/drafts/\[DRAFT\\_ID\]](http://www[XXX].slip-emailing.com/sliprestful/bases/[DATABASE_API_ID]/drafts/[DRAFT_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Draft edited

400 – Validation errors

403 - Access forbidden

404 – Database not found

405 - Invalid HTTP method

Response->status: ok or error

Response->message: brief description with the execution results

Response->result: draft Id

Response-> w3cValidatorResults: W3C validation results

Response-> spamTestResults: Spam test validation results

### ***3.12 – Methods for Schedules***

#### ***3.12.1 Schedule single message***

Schedule a single message

Parameter	Type	Description
-----------	------	-------------



<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_ID</b>	Integer	Draft identifier
<b>DATETIME (HTTP PUT method) (*)</b>	String	Date and time to send the campaign
<b>DATEFORMAT</b>	String	Format of datetime parameter. The default value for this parameter is 'yyyy-mm-dd'
<b>TRACKINGDOMAIN (HTTP PUT method) (*)</b>	String	Domain for tracking. It must exist in the database
<b>SENDERNAME (HTTP PUT method) (*)</b>	String	It must exist in the database
<b>SENDERDOMAIN (HTTP PUT method) (*)</b>	String	It must exist in the database
<b>SEGMENTATIONID (HTTP PUT method) (*)</b>	Integer	Segmentation identifier. It must exist in the database
<b>LIMITATION (HTTP PUT method)</b>	Integer	Maximum number of emails to be sent
<b>REPLYTO (HTTP PUT method) (*)</b>	String	Valid email that must exist in the database
<b>TIMEZONE (HTTP PUT method)</b>	String	Valid time zone string
<b>HEADERID (HTTP PUT method)</b>	Integer	Header identifier, to be added in the top of the campaign
<b>FOOTERID (HTTP PUT method)</b>	Integer	Footer identifier, to be added at the bottom of the campaign
<b>SUBJECT (HTTP PUT method)</b>	String	Campaign subject
<b>NAME (HTTP PUT method)</b>	String	Campaign name
<b>UTMCAMPAIGN (HTTP PUT method)</b>	String	Google analytics utm_campaign parameter
<b>UTMMEDIUM (HTTP PUT method)</b>	String	Google analytics utm_medium parameter
<b>UTMSOURCE (HTTP PUT method)</b>	String	Google analytics utm_source parameter
<b>UTMTERM (HTTP PUT method)</b>	String	Google analytics utm_term parameter
<b>UTMCONTENT (HTTP PUT method)</b>	String	Google analytics utm_content parameter
<b>INTELLIGENTDELIVERY (HTTP</b>	String	Allowed intelligent delivery



<b>PUT method)</b>	mode
--------------------	------

(\*) Mandatory fields

**Http method:** PUT

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/draft/\[DRAFT\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/draft/[DRAFT_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

201 - Schedule created

400 - General error in schedule generation

403 - Access forbidden

404 - Database or transactional message not found

405 - Invalid HTTP method

Response->status string with: ok or error

Response->message: brief description with the execution results

Response-result-> empty or generated schedule Id

### 3.12.2 Schedule transactional

Schedule a transactional message

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_ID</b>	Integer	Draft identifier
<b>TRACKINGDOMAIN (HTTP PUT method) (*)</b>	String	Domain for tracking. It must exists in the database
<b>SENDERNAME (HTTP PUT method) (*)</b>	String	It must exists in the database
<b>SENDERDOMAIN (HTTP PUT method) (*)</b>	String	It must exists in the database



<b>REPLYTO (HTTP PUT method) (*)</b>	String	Valid email that must exists in the database
<b>HEADERID (HTTP PUT method)</b>	Integer	Header identifier, to be added in the top of the campaign
<b>FOOTERID (HTTP PUT method)</b>	Integer	Footer identifier, to be added at the bottom of the campaign
<b>SUBJECT (HTTP PUT method)</b>	String	Campaign subject
<b>NAME (HTTP PUT method)</b>	String	Campaign name
<b>UTMCAMPAIGN (HTTP PUT method)</b>	String	Google analytics utm_campaign parameter
<b>UTMMEDIUM (HTTP PUT method)</b>	String	Google analytics utm_medium parameter
<b>UTMSOURCE (HTTP PUT method)</b>	String	Google analytics utm_source parameter
<b>UTMTERM (HTTP PUT method)</b>	String	Google analytics utm_term parameter
<b>UTMCONTENT (HTTP PUT method)</b>	String	Google analytics utm_content parameter

(\*) Mandatory fields

**Http method:** PUT

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/transactional/\[DRAFT\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/transactional/[DRAFT_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 201 - Schedule created
- 400 - General error in schedule generation
- 403 - Access forbidden
- 404 - Database or transactional message not found
- 405 - Invalid HTTP method



Response->status string with: ok or error

Response->message: brief description with the execution results

Response-result-> empty or generated schedule Id

### 3.12.3 Get schedule info by Id

This action returns the information for the selected schedule

#### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SCHEDULE_ID</b>	Integer	Schedule identifier

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_ID])

**Response format:** json

#### **Response Object detail:**

Response->statusCode:

200 –Schedule found

403 – Access forbidden

404 – Schedule or database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->result: array with schedule data if exists, empty array if not.

Response->message: brief description with the execution results



#### 3.12.4 Get schedule info by tag

This action returns the information for the selected schedule

##### Parameters:

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SCHEDULE_TAG</b>	String	Schedule identifier

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_TAG\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_TAG])

**Response format:** json

##### Response Object detail:

Response->statusCode:

- 200 – Schedule found
- 403 – Access forbidden
- 404 – Schedule or database not found
- 405 – Invalid HTTP method

Response->status: string with: ok or error

Response->result: array with schedule data if exists, empty array if not.

Response->message: brief description with the execution results

#### 3.12.5 Get list of schedules

This action returns all schedules in the database. The results are paginated, and the maximum number of schedules per page is 1000.

##### Parameters:



Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>P (HTTP GET parameter)</b>	Integer	page

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules? p=1](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules? p=1)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Schedule list found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of schedules

Response->result->total: integer with the total number of schedules

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results





### 3.12.6 Get list of schedules with filters

This action returns all schedules in the database that match with all filters received. The results are paginated, and the maximum number of schedules per page is 1000.

#### Parameters:

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>P (HTTP GET parameter)</b>	Integer	Page of results
<b>STATUS (HTTP GET parameter)</b>	String	Allowed values: -OK -SENDING -PENDING -OPEN_TRIGGER -CLICK_TRIGGER -BIRTHDAY_TRIGGER -TRANSACTIONAL -STOP -OPEN_TRIGGER_STOP -CLICK_TRIGGER_STOP -TRANSACTIONAL_STOP -BIRTHDAY_TRIGGER_STOP
<b>TIMEZONE (HTTP GET parameter)</b>	String	Valid time zone string
<b>FROMSCHEDULEDT (HTTP GET parameter)</b>	String	From schedule date time
<b>TOSCHEDULEDT (HTTP GET parameter)</b>	String	To schedule date time
<b>DATEFORMAT</b>	String	Format of datetime parameter. The default value for this parameter is 'yyyy-mm-dd'

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules?status=transactional&dateformat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules?status=transactional&dateformat=yyyy-mm-dd)

**Response format:** json

**Response Object detail:**



Response->statusCode:

200 – Schedule list found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of schedules

Response->result->total: integer with the total number of schedules

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### 3.12.7 Send Bat to email

This function sends a BAT for the selected draft to the selected email.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_ID</b>	Integer	Draft identifier
<b>EMAIL</b>	String	Recipient email
<b>HEADERID (HTTP PUT method)</b>	Integer	Header identifier, to be added in the top of the campaign
<b>FOOTERID (HTTP PUT method)</b>	Integer	Footer identifier, to be added at the bottom of the



campaign

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/bat/\[DRAFT\\_ID\]/\[EMAIL\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/bat/[DRAFT_ID]/[EMAIL])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – BAT has been sent
- 400 – Email validation error
- 403 - Access forbidden
- 404 - Database or draft not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results.

Response->result: schedule id that has been generated.

### 3.12.8 Send Bat to List of Contacts

This function sends a BAT for the selected draft to the selected email.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_ID</b>	Integer	Draft identifier
<b>LIST_ID</b>	Integer	List identifier
<b>HEADERID (HTTP PUT method)</b>	Integer	Header identifier, to be added in the top of the campaign



<b>FOOTERID (HTTP PUT method)</b>	Integer	Footer identifier, to be added at the bottom of the campaign
-----------------------------------	---------	--

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/bat/\[DRAFT\\_ID\]/\[LIST\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/bat/[DRAFT_ID]/[LIST_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – BAT has been sent
- 400 – Email validation error
- 403 - Access forbidden
- 404 – Database, list or draft not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results.

Response->result: schedule id that has been generated.

### 3.12.9 Stop schedule

This function stops a schedule. Only campaigns in status PENDING or SENDING can be stopped. Trigger and transactional messages can be stopped too if they are actives (status equal to TRANSACTIONAL, OPEN\_TRIGGER, CLICK\_TRIGGER or BIRTHDAY\_TRIGGER).

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SCHEDULE_ID or SCHEDULE_TAG</b>	Integer / string	Schedule identifier



**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_ID\]/stop](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_ID]/stop)

Or

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_TAG\]/stop](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_TAG]/stop)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – Schedule has been stopped
- 400 – Schedule with invalid status to be stopped
- 403 - Access forbidden
- 404 – Schedule not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results.

Response->result: empty array.

#### **3.12.10 Restart schedule**

This function restarts a schedule that is stopped. Only campaigns in status STOP can be restarted. Trigger and transactional messages can be restarted too if they are stopped (status equal to TRANSACTIONAL\_STOP, OPEN\_TRIGGER\_STOP, CLICK\_TRIGGER\_STOP or BIRTHDAY\_TRIGGER\_STOP).

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SCHEDULE_ID or SCHEDULE_TAG</b>	Integer / String	Schedule identifier

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_ID\]/start](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_ID]/start)

Or

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_TAG\]/start](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_TAG]/start)

**Response format:** json**Response Object detail:**

Response->statusCode:

- 200 – Schedule has been restarted
- 400 – Schedule with invalid status to be started
- 403 - Access forbidden
- 404 – Schedule not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results.

Response->result: empty array



### 3.12.11 Remove schedule

This function removes a schedule. Only schedules in status PENDING or STOP can be removed. Trigger and transactional messages cannot be removed (see previous point to know how to stop trigger and transactional messages).

#### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SCHEDULE_ID or SCHEDULE_TAG</b>	Integer / string	Schedule identifier

**Http method:** DELETE

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_ID])

Or

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/schedules/\[SCHEDULE\\_TAG\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/schedules/[SCHEDULE_TAG])

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – Schedule has been removed
- 400 – Schedule with invalid status to be removed
- 403 - Access forbidden
- 404 – Schedule not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results.



Response->result: empty array

### 3.13 - Methods for Sender Names

#### 3.13.1 Get Sender Names

This function returns a list with all sender names for the selected database.

##### Parameters:

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/sender\\_names](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/sender_names)

**Response format:** json

##### Response Object detail:

Response->statusCode:

200 – Sender names list found

403 – Access forbidden

404 – Database not found or no sender names for the selected database

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of sender names

Response->result->total: integer with the total number of sender names

Response->result->page: number of current page





Response->result->totalPages: total number of pages

### 3.13.2 Get sender name by id

This action returns the selected sender name for the selected database

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SENDER_NAME_ID</b>	INTEGER	Sender name identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/sender\\_names/\[SENDER\\_NAME\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/sender_names/[SENDER_NAME_ID])

**Response format:** json

Response->statusCode:

- 200 – Sender name found
- 403 - Access forbidden
- 404 - Database or sender name not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected sender name. The properties for the sender name are:

- id: Integer
- name: String
- creation: Datetime (referred to returned time zone)
- timezone: String
- isRemoved: Boolean
- isDefault: Boolean



Response->result->total: number of sender names

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.13.3 Get default sender name

This action returns the default sender name for the selected database

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/sender\\_names/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/sender_names/default)

**Response format:** json

Response->statusCode:

- 200 – Sender name found
- 403 - Access forbidden
- 404 - Database or sender name not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected sender name. The properties for the sender name are:

- id: Integer
- name: String
- creation: Datetime (referred to returned time zone)



- timezone: String
- isRemoved: Boolean
- isDefault: Boolean

Response->result->total: number of sender names

Response->result->page: number of current page

Response->result->totalPages: total number of pages

#### ***3.13.4 Set default sender name***

Set the selected sender name as default. Only one default sender name is allowed by database.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SENDER_NAME_ID</b>	INTEGER	Sender name identifier

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/sender\\_names/\[SENDER\\_NAME\\_ID\]/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/sender_names/[SENDER_NAME_ID]/default)

**Response format:** json

Response->statusCode:

200 – Sender name set to default

403 - Access forbidden

404 - Database or sender name not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results



Response->result: empty

### 3.13.5 Create new sender name

Add new sender name to the database

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>SENDER_NAME</b>	String	Sender name

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/sender\\_names/\[SENDER\\_NAME\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/sender_names/[SENDER_NAME])

**Response format:** json

Response->statusCode:

200 – Sender name successfully created

400 – Invalid value for Sender Name

403 - Access forbidden

404 - Database not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: Sender name Id



### 3.14 - Methods for Statistics

#### 3.14.1 Get Statistics Summary for single campaigns

Get a general statistics summary for the campaigns that match with the filters. If none criterion is passed, returns the statistics for all campaigns in the selected database

##### Parameters:

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>P (HTTP GET parameter)</b>	Integer	Page of results
<b>TIMEZONE (HTTP GET parameter)</b>	String	Valid time zone string
<b>FROMDATE (HTTP GET parameter)</b>	String	From date time
<b>TODATE (HTTP GET parameter)</b>	String	To date time
<b>DATEFORMAT</b>	String	Format of datetime parameter. The default value for this parameter is 'yyyy-mm-dd'

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/campaign?p=1&fromDate=2012-01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid&dateFormat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/campaign?p=1&fromDate=2012-01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid&dateFormat=yyyy-mm-dd)

**Response format:** json

##### Response Object detail:

Response->statusCode:

- 200 – Statistics summary found
- 400 - Invalid parameter
- 403 – Access forbidden
- 404 – Database not found
- 405 – Invalid HTTP method

Response->status: string with: ok or error



Response->message: brief description with the execution results

Response->result->data: Array of schedules statistics

Response->result->total: integer with the total number of schedules

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

### 3.14.2 Get Statistics Summary for triggers

Get a general statistics summary for the triggers that match with the filters. If none criterion is passed, returns the statistics for all of them in the selected database

#### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>P (HTTP GET parameter)</b>	Integer	Page of results
<b>TIMEZONE (HTTP GET parameter)</b>	String	Valid time zone string
<b>FROMDATE (HTTP GET parameter)</b>	String	From date time
<b>TODATE (HTTP GET parameter)</b>	String	To date time
<b>DATEFORMAT</b>	String	Format of datetime parameter. The default value for this parameter is 'yyyy-mm-dd'

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/trigger?p=1&fromDate=2012-01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid&dateformat=yyyy-mm-dd](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/trigger?p=1&fromDate=2012-01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid&dateformat=yyyy-mm-dd)

**Response format:** json

**Response Object detail:**

Response->statusCode:

- 200 – Statistics summary found
- 400 - Invalid parameter
- 403 – Access forbidden
- 404 – Database not found
- 405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of schedules statistics

Response->result->total: integer with the total number of schedules

Response->result->page: number of current page

Response->result->totalPages: total number of pages

Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

**3.14.3 Get Statistics Summary for transactional messages**

Get a general statistics summary for the transactional messages that match with the filters. If none criterion is passed, returns the statistics for all of them in the selected database

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>P (HTTP GET parameter)</b>	Integer	Page of results
<b>TIMEZONE (HTTP GET parameter)</b>	String	Valid time zone string



<b>FROMDATE (HTTP GET parameter)</b>	String	From date time
<b>TODATE (HTTP GET parameter)</b>	String	To date time
<b>DATEFORMAT</b>	String	Format of datetime parameter. The default value for this parameter is 'yyyy-mm-dd'

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/transactional?p=1&fromDate=2012-01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/transactional?p=1&fromDate=2012-01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid)

**Response format:** json

#### **Response Object detail:**

Response->statusCode:

- 200 – Statistics summary found
- 400 - Invalid parameter
- 403 – Access forbidden
- 404 – Database not found
- 405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of schedules statistics

Response->result->total: integer with the total number of schedules

Response->result->page: number of current page

Response->result->totalPages: total number of pages





Response->result->links->urlNextPage: Url for the next page of results

Response->result->links->urlPreviousPage: Url for the previous page of results

#### **3.14.4 Get Statistics Summary for single campaign (by schedule Id)**

Get statistics summary for the selected campaign

##### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>CAMPAIGN_ID</b>	Integer	Campaign identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/campaign/\[CAMPAIGN\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/campaign/[CAMPAIGN_ID])

**Response format:** json

##### **Response Object detail:**

Response->statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Object with the statistics for the selected campaign



### 3.14.5 Get Statistics Summary for single campaign (by schedule Tag)

Get statistics summary for the selected campaign

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>CAMPAIGN_TAG</b>	String	Campaign tag

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/campaign/\[CAMPAIGN\\_TAG\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/campaign/[CAMPAIGN_TAG])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Object with the statistics for the selected campaign



### 3.14.6 Get Statistics Summary for transactional message (by schedule Id)

Get statistics summary for the selected transactional message

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRANSACTIONAL_ID</b>	Integer	Transactional message identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/transactional/\[TRANSACTIONAL\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/transactional/[TRANSACTIONAL_ID])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Object with the statistics for the selected transactional message



### 3.14.7 Get Statistics Summary for transactional message (by schedule Tag)

Get statistics summary for the selected transactional message

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRANSACTIONAL_TAG</b>	Integer	Transactional message identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/transactional/\[TRANSACTIONAL\\_TAG\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/transactional/[TRANSACTIONAL_TAG])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Object with the statistics for the selected transactional message

### 3.14.8 Get Statistics Summary for trigger (by schedule Id)

Get statistics summary for the selected trigger message

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRIGGER_ID</b>	Integer	Trigger message identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/trigger/\[TRIGGER\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/trigger/[TRIGGER_ID])

**Response format:** json**Response Object detail:**

Response-&gt;statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response-&gt;status: string with: ok or error

Response-&gt;message: brief description with the execution results

Response-&gt;result-&gt;data: Object with the statistics for the selected trigger

**3.14.9 Get Statistics Summary for trigger (by schedule Tag)**

Get statistics summary for the selected trigger message

**Parameters:**

Parameter	Type	Description
-----------	------	-------------



<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRIGGER_TAG</b>	String	Trigger message tag

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/trigger/\[TRIGGER\\_TAG\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/trigger/[TRIGGER_TAG])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Object with the statistics for the selected trigger

#### 3.14.10 Get statistics by origin

This method returns a summary of the results grouped by origin.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>TIMEZONE (HTTP GET parameter)</b>	String	Valid time zone string
<b>FROMDATE (HTTP GET parameter)</b>	String	From date time



<b>TODATE (HTTP GET parameter)</b>	String	To date time
------------------------------------	--------	--------------

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/stats/origin? fromDate=2013-05-01+00:00:01&toDate=2013-05-30+23:59:59&timeZone=Europe/Madrid](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/stats/origin? fromDate=2013-05-01+00:00:01&toDate=2013-05-30+23:59:59&timeZone=Europe/Madrid)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Statistics summary found

400 - Invalid parameter

403 – Access forbidden

404 – Database not found

405 – Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result: Object with the statistics for each origin

### ***3.15 Methods for headers***

#### ***3.15.1 Get headers***

This action returns the headers for the selected database.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier



**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/headers](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/headers)

**Response format:** json

Response->statusCode:

200 – Headers found

403 - Access forbidden

404 - Database or headers not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of headers. The properties for the each header are:

- id: Integer
- name: String
- base: String
- creation: Datetime (referred to timezone)
- timezone: String
- html: String
- isDefault: Boolean

Response->result->total: number of headers

Response->result->page: number of current page

Response->result->totalPages: total number of pages





### 3.15.2 Get header by id

This action returns the selected header

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>HEADER_ID</b>	INTEGER	Header identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/headers/\[HEADER\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/headers/[HEADER_ID])

**Response format:** json

Response->statusCode:

200 – Header found

403 - Access forbidden

404 - Database or header not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected header. The properties for the header are:

- id: Integer
- name: String
- base: String
- creation: Datetime (referred to timezone)
- timezone: String
- html: String
- isDefault: Boolean



Response->result->total: number of headers

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.15.3 Get default header

This action returns the default header for the selected database

#### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/headers/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/headers/default)

**Response format:** json

Response->statusCode:

200 – Header found

403 - Access forbidden

404 - Database or header not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected header. The properties for the header are:



- id: Integer
- name: String
- base: String
- creation: Datetime (referred to timezone)
- timezone: String
- html: String
- isDefault: Boolean

Response->result->total: number of headers

Response->result->page: number of current page

Response->result->totalPages: total number of pages

#### 3.15.4 Set default header

Set the selected header as default. Only one default header is allowed by database.

#### Parameters:

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>HEADER_ID</b>	INTEGER	Header identifier

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/headers/\[HEADER\\_ID\]/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/headers/[HEADER_ID]/default)

**Response format:** json

Response->statusCode:

200 – Header set to default

403 - Access forbidden

404 - Database or header not found



405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: empty

### ***3.16 Methods for footers***

#### **3.16.1 Get footers**

This action returns the footers for the selected database.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/footers](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/footers)

**Response format:** json

Response->statusCode:

200 - Footers found

403 - Access forbidden

404 - Database or footers not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of footers. The properties for the each footer are:



- id: Integer
- name: String
- base: String
- creation: Datetime (referred to timezone)
- timezone: String
- html: String
- isDefault: Boolean

Response->result->total: number of footers

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.16.2 Get footer by id

This action returns the selected footer

#### Parameters:

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier
FOOTER_ID	INTEGER	Footer identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/footers/\[FOOTER\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/footers/[FOOTER_ID])

**Response format:** json

Response->statusCode:

200 – Footer found

403 - Access forbidden

404 - Database or footer not found



#### 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected footer. The properties for the footer are:

- id: Integer
- name: String
- base: String
- creation: Datetime (referred to timezone)
- timezone: String
- html: String
- isDefault: Boolean

Response->result->total: number of footers

Response->result->page: number of current page

Response->result->totalPages: total number of pages

#### 3.16.3 Get default footer

This action returns the default footer for the selected database

##### Parameters:

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier

Http method: GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/footers/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/footers/default)

Response format: json



Response->statusCode:

200 – Footer found

403 - Access forbidden

404 - Database or footer not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected footer. The properties for the footer are:

- id: Integer
- name: String
  
- base: String
- creation: Datetime (referred to timezone)
- timezone: String
- html: String
- isDefault: Boolean

Response->result->total: number of footers

Response->result->page: number of current page

Response->result->totalPages: total number of pages

#### 3.16.4 Set default footer

Set the selected footer as default. Only one default footer is allowed by database.

**Parameters:**

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier



<b>FOOTER_ID</b>	INTEGER	Footer identifier
------------------	---------	-------------------

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/footers/\[FOOTER\\_ID\]/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/footers/[FOOTER_ID]/default)

**Response format:** json

Response->statusCode:

200 – Footer set to default

403 - Access forbidden

404 - Database or footer not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: empty

### ***3.17 Methods for Replies to***

#### ***3.17.1 Get Replies to***

This action returns the “replies to” for the selected database.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/replies\\_to](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/replies_to)





**Response format:** json

Response->statusCode:

200 – Replies to found

403 - Access forbidden

404 - Database or “replies to” not found

405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of “replies to”. The properties for the each reply to are:

- id: Integer
- name: String
- creation: Datetime (referred to timezone)
- timezone: String
- isDefault: Boolean

Response->result->total: number of “replies to”

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.17.2 Set default reply to

Set the selected reply to as default. Only one default “reply to” is allowed by database.

**Parameters:**

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier
REPLY_TO_ID	INTEGER	Reply to identifier



**Http method:** PUT

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/replies to/\[REPLY\\_TO\\_ID\]/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/replies to/[REPLY_TO_ID]/default)

**Response format:** json

Response->statusCode:

- 200 – Reply to set to default
- 403 - Access forbidden
- 404 - Database or tracking domain not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: empty

### **3.17.3 Get reply to by id**

This action returns the selected reply to

Parameters:

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier
REPLY_TO_ID	INTEGER	Reply to identifier

**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/replies to/\[REPLY\\_TO\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/replies to/[REPLY_TO_ID])

**Response format:** json

Response->statusCode:



200 – Reply to found

403 - Access forbidden

404 - Database or reply to not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected reply to. The properties for the reply to are:

- id: Integer
- creation: Datetime (referred to timezone)
- timezone: String
- isDefault: Boolean
- name: String

Response->result->total: number of returned replies to

Response->result->page: number of current page

Response->result->totalPages: total number of pages

#### 3.17.4 Get default reply to

This action returns the default reply to in the database

Parameters:

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier

Http method: GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/replies\\_to/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/replies_to/default)



Response format: json

Response->statusCode:

- 200 – Reply to found
- 403 - Access forbidden
- 404 - Database or reply to not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with default “reply to”. The properties for the reply to are:

- id: Integer
- creation: Datetime (referred to timezone)
- timezone: String
- isDefault: Boolean
- name: String

Response->result->total: number of returned replies to

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### ***3.18 Methods for Tracking Domains***

#### **3.18.1 Get Tracking Domains**

This action returns all the “tracking domains” for the selected database.

**Parameters:**

Parameter	Type	Description
-----------	------	-------------



<b>DATABASE_API_ID</b>	String	Database identifier
------------------------	--------	---------------------

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/tracking\\_domains](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/tracking_domains)

**Response format:** json

Response->statusCode:

- 200 – Tracking domains found
- 403 - Access forbidden
- 404 - Database or tracking domains not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of “tracking domains”. The properties for the each reply to are:

- id: Integer
- domain: String
- isDefault: Boolean

Response->result->total: number of “tracking domains” returned

Response->result->page: current page value

Response->result->totalPages: total number of pages

### 3.18.2 Get tracking domain by id

This action returns the selected tracking domain

**Parameters:**

Parameter	Type	Description
-----------	------	-------------



<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRACKING_DOMAIN_ID</b>	INTEGER	Tracking domain identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/tracking\\_domains/\[TRACKING\\_DOMAIN\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/tracking_domains/[TRACKING_DOMAIN_ID])

**Response format:** json

Response->statusCode:

- 200 - Tracking domain found
- 403 - Access forbidden
- 404 - Database or tracking domain not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected tracking domain. The properties for the tracking domain are:

- id: Integer
- domain: String
- isDefault: Boolean

Response->result->total: number of returned tracking domains

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.18.3 Get default tracking domain

This action returns the default tracking domain for the selected database

**Parameters:**



Parameter	Type	Description
DATABASE_API_ID	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/tracking\\_domain/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/tracking_domain/default)

**Response format:** json

Response->statusCode:

- 200 – Tracking domain found
- 403 - Access forbidden
- 404 - Database or tracking domain not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected tracking domain. The properties for the tracking domain are:

- id: Integer
- domain: String
- isDefault: Boolean

Response->result->total: number of tracking domains

Response->result->page: number of current page

Response->result->totalPages: total number of pages

#### **3.18.4 Set default tracking domain**

Set the selected tracking domain as default. Only one default tracking domain is allowed by database.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>TRACKING_DOMAIN_ID</b>	INTEGER	Tracking domain identifier

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/tracking\\_domains/\[TRACKING\\_DOMAIN\\_ID\]/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/tracking_domains/[TRACKING_DOMAIN_ID]/default)

**Response format:** json

Response->statusCode:

200 – Tracking domain set to default

403 - Access forbidden

404 - Database or tracking domain not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: empty

### ***3.19 Methods for Domains***

#### ***3.19.1 Get Domains***

This action returns all the “domains” for the selected database.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET





URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/domains](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/domains)

**Response format:** json

Response->statusCode:

- 200 – Domains found
- 403 - Access forbidden
- 404 - Database or tracking domains not found
- 405 - Invalid HTTP method

Response->status: string with: ok or error

Response->message: brief description with the execution results

Response->result->data: Array of “domains”. The properties for the each reply to are:

- id: integer value
- name: string
- senderId: integer
- isDefault: boolean

Response->result->total: number of “domains” returned

Response->result->page: current page value

Response->result->totalPages: total number of pages

### 3.19.2 Get domain by id

This action returns the selected domain

**Parameters:**

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier
DOMAIN_ID	INTEGER	Domain identifier

**Http method:** GET



URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/domains/\[DOMAIN\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/domains/[DOMAIN_ID])

**Response format:** json

Response->statusCode:

- 200 – Domain found
- 403 - Access forbidden
- 404 - Database or domain not found
- 405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected domain. The properties for the domain are:

- id: Integer
- name: String
- senderId: Integer
- isDefault: Boolean

Response->result->total: number of returned domains

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.19.3 Get default domain

This action returns the default domain for the selected database

**Parameters:**

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier



**Http method:** GET

**URL:** [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/domain/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/domain/default)

**Response format:** json

Response->statusCode:

200 – Domain found

403 - Access forbidden

404 - Database or domain not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result->data: One element array with the selected domain. The properties for the domain are:

- id: Integer
- name: String
- senderId: Integer
- isDefault: Boolean

Response->result->total: number of domains

Response->result->page: number of current page

Response->result->totalPages: total number of pages

### 3.19.4 Set default domain

Set the selected domain as default. Only one default domain is allowed by database.

**Parameters:**

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier



<b>DOMAIN_ID</b>	INTEGER	Domain identifier
------------------	---------	-------------------

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/domains/\[DOMAIN\\_ID\]/default](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/domains/[DOMAIN_ID]/default)

**Response format:** json

Response->statusCode:

200 – Domain set to default

403 - Access forbidden

404 - Database or Domain not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: empty

## 3.20 Methods for Split Sendouts

### 3.20.1 Create new split sendout

This method generates a split sendout for the received list of drafts.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>DRAFT_LIST (*)</b>	String	Separated by commas string with the list of draftIds
<b>NAME (HTTP PUT method) (*)</b>	String	Name for the split
<b>DATETIME (HTTP PUT method) (*)</b>	String	Date and time to send the campaign



# LIVEDATA

Smart Technology for Data in Motion

<b>DATEFORMAT</b>	String	Format of datetime parameter. The default value for this parameter is 'yyyy-mm-dd'
<b>TRACKINGDOMAIN (HTTP PUT method) (*)</b>	String	Domain for tracking. It must exist in the database
<b>SENDERNAME (HTTP PUT method) (*)</b>	String	It must exist in the database
<b>SENDERDOMAIN (HTTP PUT method) (*)</b>	String	It must exist in the database
<b>SEGMENTATIONID (HTTP PUT method) (*)</b>	Integer	Segmentation identifier. It must exist in the database
<b>PERCENTAGE (HTTP PUT method) (*)</b>	Integer	Over the total contacts in segmentation
<b>REPLYTO (HTTP PUT method) (*)</b>	String	Valid email that must exist in the database
<b>TIMEZONE (HTTP PUT method)</b>	String	Valid time zone string
<b>HEADERID (HTTP PUT method)</b>	Integer	Header identifier, to be added in the top of the campaign
<b>FOOTERID (HTTP PUT method)</b>	Integer	Footer identifier, to be added at the bottom of the campaign
<b>WINNER TYPE (HTTP PUT method)</b>	String (valid values: open, click or manually)	Sets the way the test winner is chosen, after the time mentioned in the parameter WINNERAFTER.
<b>WINNERAFTER (HTTP PUT method)</b>	INTEGER	Set the number of hours the application will wait before choose the winner for the split. If WINNERTYPE is set to "manually" this parameter must not be set. If WINNERTYPE is set to "click" or "open", after the established period, a new schedule will be automatically generated in order to send the winner campaign to the rest of the contacts in the segmentation



(\*) Mandatory fields

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/split\\_sendouts](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/split_sendouts)

**Response format:** json

**Response Object detail:**

Response->statusCode:

201 - Split created

400 - General error in split generation, like validation errors

403 - Access forbidden

404 – Database not found

405 - Invalid HTTP method

Response->status string with: ok or error

Response->message: brief description with the execution results

Response->result: empty or generated split Id

### ***3.21 – Methods for intelligent delivery modes***

#### ***3.21.1 Get Intelligent Delivery mode***

This action returns a list of allowed intelligent delivery modes when scheduling a campaign.

**Parameters:**

Parameter	Type	Description
DATABASE_API_ID	String	Database identifier



**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/intelligent\\_delivery](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/intelligent_delivery)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 – Intelligent delivery methods found

403 - Access forbidden

404 – Database not found or intelligent delivery modes not available

405 - Invalid HTTP method

Response->status string with: ok or error

Response->message: brief description with the execution results

Response-> result->data: Array of different intelligent delivery modes

## ***3.22 Methods for Attributes***

### ***3.22.1 Get Attributes***

This action returns a list of allowed attributes for contacts in database.

**Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/attributes](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/attributes)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Attributes found

403 - Access forbidden

404 – Database not found or no attributes for the selected database

405 - Invalid HTTP method

Response->status string with: ok or error

Response->message: brief description with the execution results

Response-> result->data: Array of different attributes

Response->result->total: number of domains

Response->result->page: number of current page

Response->result->totalPages: total number of pages

**3.22.2 Get attribute by id**

This action returns the selected attribute for the selected database

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>ATTRIBUTE_ID</b>	INTEGER	Sender name identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/attributes/\[ATTRIBUTE\\_ID\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/attributes/[ATTRIBUTE_ID])

**Response format:** json

Response->statusCode:

200 - Attribute found





403 - Access forbidden

404 - Database or attribute not found

405 - Invalid HTTP method

Response->status: string with ok or error

Response->message: brief description with the execution results

Response->result: Array with the properties for the selected attribute:

- id: Integer
- name: String
- internalName: String
- creation: Datetime (referred to returned time zone)
- timezone: String
- isCustom: Boolean
- dataType: String

### 3.22.3 Get Allowed Attribute's Data types

This action returns a list of allowed data types for attributes.

#### **Parameters:**

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier

**Http method:** GET

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/attributes/customs/data\\_types](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/attributes/customs/data_types)

**Response format:** json

#### **Response Object detail:**

Response->statusCode:

200 - Data Types found



403 - Access forbidden

404 – Database not found or no attribute's data types defined

405 - Invalid HTTP method

Response->status string with: ok or error

Response->message: brief description with the execution results

Response-> result->data: Array of different attributes

### 3.22.4 Create new Custom Attribute

Generate a new custom attribute.

Parameter	Type	Description
<b>DATABASE_API_ID (*)</b>	String	Database identifier
<b>NAME (*)</b>	String	List name. Must be unique
<b>TYPE (*)</b>	String	Allowed types: string, integer, boolean, date and datetime. (in lowercase)

(\*) Mandatory fields

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/attributes/customs](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/attributes/customs)

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Custom attribute created

400 – Error while creating attribute

403 - Access forbidden



404 - Database not found

405 - Invalid HTTP method

500 - Internal server error

Response->status: ok or error

Response->message: Brief description with the execution results

Response->result: Attribute Id

### 3.23 Methods for Scenario Manager

#### 3.23.1 Add contact to program

Add one contact in database to active program in scenario manager.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>PROGRAM_ID</b>	Integer	List identifier
<b>EMAIL</b>	String or Integer	Existing email in database or contact Id

**Http method:** PUT

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/scenario/programs/\[PROGRAM\\_ID\]/\[EMAIL\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/scenario/programs/[PROGRAM_ID]/[EMAIL])

**Response format:** json

**Response Object detail:**

Response->statusCode:

201 - Contact added to program

400 - Error adding contact

403 - Access forbidden



404 - Program or contact not found

405 - Invalid HTTP method

500 - Internal server error

Response->status: ok or error

Response->message: Brief description with the execution results

Response->result: Empty array

### 3.23.2 Remove contact from program

Remove one contact in database from active program in scenario manager.

Parameter	Type	Description
<b>DATABASE_API_ID</b>	String	Database identifier
<b>PROGRAM_ID</b>	Integer	List identifier
<b>EMAIL</b>	String or Integer	Existing email in database or contact Id

**Http method:** DELETE

URL: [http://www\[XXX\].livedata-emailing.com/sliprestful/bases/\[DATABASE\\_API\\_ID\]/scenario/programs/\[PROGRAM\\_ID\]/\[EMAIL\]](http://www[XXX].livedata-emailing.com/sliprestful/bases/[DATABASE_API_ID]/scenario/programs/[PROGRAM_ID]/[EMAIL])

**Response format:** json

**Response Object detail:**

Response->statusCode:

200 - Contact removed from program

400 - Error removing contact

403 - Access forbidden

404 - Program or contact not found



405 - Invalid HTTP method

500 - Internal server error

Response->status: ok or error

Response->message: Brief description with the execution results

Response->result: Empty array

## 4 – Authentication

The Live Data RESTful API use WSSE (Web Service – Security extension) for authenticating the requests of the users.

The user request must contain a WSSE header similar to:

**GET /resource.html HTTP/1.1**

**Host: www.example.com**

**Authorization: WSSE profile="UsernameToken"**

**X-WSSE: UsernameToken Username="Alibaba",**

**PasswordDigest="Z2Y59TewHV6r9BWjtHLkKfUjm2k=",**

**Nonce="EFD89F06CCB28C89",**

**Created="2007-04-13T09:00:00Z"**

The username and the password can be found in the Database Settings menu of Live Data Emailing Platform.



## 5 – Examples

### Note:

The attached examples use php as client web language and the CURL library in order to execute the request and obtain the response from the web server.

### 5.1 Basic script structure (Get segmentation query string)

```
<?php
$fieldsString = '';
$ch=curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL,'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/segments/999/query?f1=EMAIL
&f2=FIRSTNAME&f3=ID');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
$result_decode = json_decode($response); // decoding the response

/**
 *
 * Generate the WSSE header for authentication
 */
function generateWSSEHeader($username, $password){
    $created = date('c');
    $nonce = substr(md5(uniqid('nonce_', true)),0,16);

    $nonce64 = base64_encode($nonce);
    $passwordDigest = base64_encode(sha1($nonce . $created .
$password, true));

    return array('X-WSSE: UsernameToken Username="' . $username .
'", PasswordDigest="' . $passwordDigest . '", Nonce="' . $nonce64
. '", Created="' . $created . '"');
```



```
}
?>
```

## 5.2 Getting contact(s)

### 5.2.1 Get contact by email

```
<?php
$fieldsString = '';
$ch=curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-contact
@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.2.2 Get contact by id

```
<?php
$fieldsString = '';

$ch=curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-
contact@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
```



```

    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    ...
    // the rest of the script is similar to the basic script

```

### 5.2.3 Get contacts

#### **A. - All contacts**

```

<?php
    $fieldsString = '';

    $ch=curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch,CURLOPT_URL,'http://www5.livedata-emailing.com
/sliprestful/bases/YOUR_API_ID/contacts');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    ...

    // the rest of the script is similar to the basic script

```

#### **B - Get contacts using filters**

```

<?php
    $fieldsString = '';
    $ch=curl_init(); // initialize curl library

```





```
// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts?firstname=John&tit
le=mr');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

#### 5.2.4 Get Unsubscribes / opens / bounces / clicks

```
<?php
$fieldsString = '';
$ch=curl_init(); // initialize curl library
// set curl options
// for unsubscribes
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/unsubscribes?fromDate=2013-
03-21+10:00:00&toDate=2013-03-
21+14:40:00&p=1&timezone=Europe/London');
// for opens
/* curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/opens?fromDate=2013-03-
21+10:00:00&toDate=2013-03-
21+14:40:00&p=1&timezone=Europe/London&schedule=123'); */
// for bounces

/* curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/bounces?fromDate=2013-03-
21+10:00:00&toDate=2013-03-21+14:40:00&p=1&timezone=Europe/London'); */
// for clicks
/* curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/clicks?fromDate=2013-03-
```



```
21+10:00:00&toDate=2013-03-
21+14:40:00&p=1&timezone=Europe/London&schedule=123');*/

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    ...
    // the rest of the script is similar to the basic script
```

### 5.3 Managing contacts (add, edit, unsubscribe, remove)

#### 5.3.1 Add / update contact to database

```
<?php
    // Contact attributes must be sent with PUT HTTP method
    $fieldsString =
'firstname=ContactName&lastname=ContactLastName&title=Mr&list_id=22';
    $ch=curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-
contact@domain.com');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process

    $headers = generateWSSEHeader('your_username',
'your_api_password');

    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
```



```

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
$result_decode = json_decode($response); // decoding the response
...
// rest of the script

```

The code for adding or updating contact is the same. If the contact does not exist, a new contact will be inserted; if the contact exists, its attributes will be modified.

### 5.3.2 Delete contact from database

```

<?php
$fieldsString = '';
$ch=curl_init(); // initialize curl library

// set curl options

curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-contact
@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your username', 'your api password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'DELETE'); // the HTTP method
must be set to DELETE
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
$result_decode = json_decode($response); // decoding the response
...
// rest of the script

```

### 5.3.3 Unsubscribe contact

```

<?php
$fieldsString = '';
$ch=curl_init(); // initialize curl library

// set curl options

```



```

    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-
contact@domain.com/unsubscription');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP method
must be set to PUT
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    $result_decode = json_decode($response); // decoding the response
    ...
    // rest of the script

```

### 5.3.4 Resubscribe contact

```

<?php
    $fieldsString = '';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-
contact@domain.com/resubscription');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP method
must be set to PUT

    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    $result_decode = json_decode($response); // decoding the response
    ...
    // rest of the script

```



### 5.3.5 Activate contact in status queued

```
<?php
    $fieldsString = '';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
    emailing.com/sliprestful/bases/YOUR_API_ID/contacts/the-
    contact@domain.com/activation');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
    'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP method
    must be set to PUT
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);

    $response = curl_exec($ch); // send the request and get the server
    response
    curl_close($ch); // close the connection
    $result_decode = json_decode($response); // decoding the response
    ...
    // rest of the script
```

## 5.4 Managing Segmentations

### 5.4.1 Get Segmentations

```
<?php
    $fieldsString = '';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
    emailing.com/sliprestful/bases/YOUR_API_ID/segments');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
    'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
```



```

    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection

```

#### **5.4.2 Generate segmentation from list of contacts**

```

<?php
    $fieldsString = 'name=My new segmentation';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/segments/contact_lists/999'
);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process

    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    ...

```

### **5.5 Managing Lists of contacts**

#### **5.5.1 Get lists of contacts from database**

```

<?php
    $fieldsString = '';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

```



```
// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
```

### 5.5.2 Get list from list Id

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists/999');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
```

### 5.5.3 Get contacts from list

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
```



```

    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists/999/contacts?
p=1');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    ...

```

#### 5.5.4 Create a new list of contacts

```

<?php

$fieldsString = '';

$fields = array('name' => urlencode('contact list from
API'),
                'description' => urlencode('my new contact list
from API'),
                'emaillist' => urlencode('the-contact-
1@doamin.com, the-contact-2@domain.com,the-contact-3@domain.com'),
                'batList' => urlencode('true'));
foreach($fields as $key => $value)
    $fieldsString .= $key.'=' . $value . '&';
rtrim($fieldsString, '&');

$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);

```





```

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...

```

### 5.5.5 Remove a list of contacts

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library
// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists/999');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'DELETE'); // the HTTP
method must be set to DELETE
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
$result_decode = json_decode($response); // decoding the response
...

```

### 5.5.6 Add contact to list

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists/999/contact/y
our_email@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

```



```

    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP method
must be set to PUT
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    $result_decode = json_decode($response); // decoding the response
    ...

```

### **5.5.7 Remove contact from list**

```

<?php
    $fieldsString = '';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/contact_lists/999/contact/y
our_email@domain.com');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    // generate the WSSE header for authentication process
    $headers = generateWSSEHeader('your_username',
'your_api_password');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'DELETE'); // the HTTP
method must be set to DELETE
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response

    curl_close($ch); // close the connection

    $result_decode = json_decode($response); // decoding the response
    ...

```

## **5.6 Custom forms**

### **5.6.1 Get HTML code for selected form**

```

<?php
    $fieldsString = '';
    $ch = curl_init(); // initialize curl library

    // set curl options
    curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/forms/999');

```



```

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script

```

## 5.7 Sender Names management

### 5.7.1 Get sender names from database

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/sender_names');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process

$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script

```

## 5.8 Sending BATs

### 5.8.1 Send BAT to email

```

<?php

```



# LIVEDATA

Smart Technology for Data in Motion

```
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules/bat/999/the-
contact@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
$fieldsString = 'footerId=99&headerId=99'; // optional parameters
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

## 5.8.2 Send BAT to list

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options

curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules/bat/99/list/999')
;
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
$fieldsString = 'footerId=99&headerId=99'; // optional parameters
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```



## 5.9 Get schedule(s) info (Single campaigns, triggers or transactionals)

### 5.9.1 Get selected schedule

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
// schedule_tag or schedule_id are allowed
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules/data_sendout_2w45
4xzffwe_99_9');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.9.2 Get a list of schedules

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
// list of all schedules (without filters)
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules');
// you can include filters adding parameters by GET in the URL
//curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules?timezone=Europe/L
ondon&fromScheduleDT=2013-03-13+17:58&toScheduleDT=2013-03-
30&status=ok'); // send filters as GET parameters
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```



```
// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.10 Schedule one campaign

```
<?php
$fieldsString = 'datetime=2013-03-28 17:25&trackingDomain=your-
tracking-domain.com&sendername=your_sender&senderDomain=sender-
domain.com&segmentationId=999&limitation=99&replyTo=the-
contact@domain.com&timezone=UTC&headerId=9&footerId=9&subject=My first
schedule from API';

$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules/draft/99');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process

$headers = generateWSSEHeader('your_username',
'your_api_password');

curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.11 Stop schedule

```
<?php
$ch = curl_init(); // initialize curl library
```



```
// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules/9999/stop');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.12 Restart stopped schedule

```
<?php
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/schedules/9999/start');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP
method must be set to PUT
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.13 Remove schedule

```
<?php
$ch = curl_init(); // initialize curl library
```



```
// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
email.com/sliprestful/bases/YOUR_API_ID/schedules/9999');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'DELETE'); // the HTTP
method must be set to DELETE
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.14 Send a transactional message

```
<?php
$fieldsString = 'email=the-contact@domain.com';
$ch = curl_init(); // initialize curl library

// set curl options

curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
email.com/sliprestful/bases/YOUR_API_ID/transactionals/9999');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, POST); // the HTTP method
must be set to POST
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```





## 5.15 Get Statistics Summary

### 5.15.1 – Get Statistics by tag or id

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
// schedule tag or schedule id are allowed
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/campaign/999');
// for triggers
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/trigger/999');
// for transactionals
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/transactional/999');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process

$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response

curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.15.2 – Get Statistics with filters

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
// schedule tag or schedule id are allowed
// for single campaigns
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/campaign?p=1&fromDate=
2012-01-15+16:00:00&toDate=2013-05-
15+17:00:00&timezone=Europe/Madrid');
```



# LIVEDATA

Smart Technology for Data in Motion

```
// for triggers
// curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/trigger?p=1&fromDate=
2012-01-15+16:00:00&toDate=2013-05-
15+17:00:00&timezone=Europe/Madrid');
// for transactionals
// curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/transactional?p=1&fro
mDate=2012-01-15+16:00:00&toDate=2013-05-
15+17:00:00&timezone=Europe/Madrid');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...

// the rest of the script is similar to the basic script
```

### 5.15.3 - Get Statistics by origin

```
<?php
$fieldsString = '';

$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/stats/origin?fromDate=2012-
01-15+16:00:00&toDate=2013-05-15+17:00:00&timezone=Europe/Madrid');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
```



```
// the rest of the script is similar to the basic script
```

## 5.16 Methods for headers

### 5.16.1 Get all headers

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/headers');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET

curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script
```

### 5.16.2 Get header by id

```
<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/headers/655');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP method
must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
```



```

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script

```

### 5.16.3 Get default header

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/headers/default');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your api password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...

// the rest of the script is similar to the basic script

```

## 5.17 Methods for footers

### 5.17.1 Get all footers

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/footers');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');

```



```

    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
    $response = curl_exec($ch); // send the request and get the server
response
    curl_close($ch); // close the connection
    ...
    // the rest of the script is similar to the basic script

```

### 5.17.2 Get footer by id

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/footers/999');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script

```

### 5.17.3 Get default footer

```

<?php
$fieldsString = '';
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/footers/default');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process

```



```

$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP method
must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
// the rest of the script is similar to the basic script

```

## 5.18 Methods for drafts

### 5.18.1 Create draft

```

<?php
$fieldsString = '';
$username     = 'your_username'; // your user
$password     = 'your_API_password'; // your apipassword
$apiId        = 'database_api_id'; // your database api id
$httpMethod   = 'PUT';

$htmlContent = '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

        <html xmlns="http://www.w3.org/1999/xhtml">
        <head>

                <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                <title>Kalkulator hipoteczny. Spelnij
marzenia o wlasnym mieszkaniu.</title>
                <style type="text/css">
                <!--
                .ReadMsgBody {width: 100%;}
                .ExternalClass {width: 100%;}
                //-->
                </style>
        </head>
        <body style="font-family: Arial; font-size:
12px;">

                <table align="center" cellspacing="0"
border="0" style="background-color: #FFFFFF;" cellpadding="0"
width="600">

                        <tr>

                                <td align="right" style="line-
height: 30px;">My new draft</td>

```



# LIVEDATA

Smart Technology for Data in Motion

```

        </tr>
    </table>
</body>
</html>';

$fields = array('name' => urlencode('My new draft (test
from API)'),
                'subject' => urlencode('Subject for my new
draft '),
                'sender' => urlencode('My sender'),
                'keywords' => urlencode('news,drafts'),
                'HtmlContent' => urlencode($htmlContent),
                'textContent' => urlencode('Text version of my
draft'),
                );

foreach($fields as $key=>$value)
    $fieldsString .= $key . '=' . $value . '&';
rtrim($fieldsString, '&');

$ch=curl_init();
curl_setopt($ch,CURLOPT_URL,'http://www5.livedata-
emailing.com/sliprestful/bases/' . $apiId . '/drafts/creation');
curl_setopt($ch,CURLOPT_POST, count($fields));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

$headers = generateWSSEHeader($username, $password);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, $httpMethod);
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);

$response = curl_exec($ch);
curl_close($ch);

function generateWSSEHeader($username, $password)
{
    $created = date('c');
    $nonce = substr(md5(uniqid('nonce_', true)),0,16);
    $nonce64 = base64_encode($nonce);
    $passwordDigest = base64_encode(sha1($nonce . $created .
$password, true));

    return array('X-WSSE: UsernameToken Username="' . $username .
'", PasswordDigest="' . $passwordDigest . '", Nonce="' . $nonce64 .
'", Created="' . $created . '"');
}

```



## 5.19 Methods for “Replies to”

### 5.19.1 Get all replies to

```
<?php
$fieldsString = '';
$username = 'your_username'; // your user
$password = 'your_API_password'; // your apipassword
$httpMethod = GET;
$ch = curl_init(); // initialize curl library

// set curl options
curl_setopt($ch, CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/replies_to');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET'); // the HTTP
method must be set to GET
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection

...
```

## 5.20 Methods for Split sendout

### 5.20.1 Create split sendout

```
<?php
$fields = array('name' => urlencode('Split from API'),
'datetime' => '2014-02-18 17:25:56',
'timezone' => 'Europe/London',
'trackingDomain' => urlencode('your-tracking-
domain.com'),
'senderName' => urlencode('your-sender-name'),
'senderDomain' => urlencode('your-sender-
domain.com'),
'segmentationId' => 999,
'replyTo' => urlencode('the-
contact@domain.com'),
'headerId' => 99
```





```

'footerId'      => 99
'draftList'     => urlencode('79,80,82'),
'percentage'    => 30
'winnerType'    => urlencode('open'), // manually ||
open || click

'winnerAfter' => 24, // time in hours
);
foreach ($fields as $key=>$value)
    $fieldsString .= $key . '=' . $value . '&';
rtrim($fieldsString, '&');

$ch=curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/split_sendouts');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username',
'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP method
must be set to PUT
curl_setopt($ch, CURLOPT_POSTFIELDS, $fieldsString);
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...

// the rest of the script is similar to the basic script

```

## 5.21 Methods for Scenario Manager

### 5.21.1 Add contact to program

```

$ch=curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/scenario/programs/999/conta
ct_xxx@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process

```



```
$headers = generateWSSEHeader('your_username', 'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT'); // the HTTP method
must be set to PUT
$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
```

### 5.21.2 Remove contact to program

```
$ch=curl_init(); // initialize curl library

// set curl options
curl_setopt($ch,CURLOPT_URL, 'http://www5.livedata-
emailing.com/sliprestful/bases/YOUR_API_ID/scenario/programs/999/conta
ct_xxx@domain.com');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// generate the WSSE header for authentication process
$headers = generateWSSEHeader('your_username', 'your_api_password');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'DELETE'); // the HTTP method
must be set to DELETE

$response = curl_exec($ch); // send the request and get the server
response
curl_close($ch); // close the connection
...
```