

***e-shot***<sup>™</sup>

# Rest API Guide

V1.18 July 2020

# Contents

<b>Introduction .....</b>	<b>5</b>
Purpose .....	5
Audience .....	5
Background .....	6
<b>Getting Started .....</b>	<b>7</b>
Overview .....	7
Quick Start .....	7
Data Reads .....	8
Data Modifications .....	9
Authentication .....	9
Token Authentication .....	9
Testing requests .....	10
<b>Settings .....</b>	<b>11</b>
System Wide Settings .....	11
Formatting .....	11
Header .....	11
Accept .....	11
Content-Type .....	12
Querystring .....	12
<b>OData HTTP GET Functionality .....</b>	<b>13</b>
Resource Paths .....	13
Pathing Structure .....	13
Query Options .....	14
Expansions .....	14
Filter .....	15
Orderby .....	19
Select .....	19
Top .....	19
Skip .....	20
Paging .....	20
Count .....	20
<b>Custom Requests .....</b>	<b>22</b>
Login .....	22

Imports.....	23
Request Parameters.....	24
Behaviour.....	25
Import Progress.....	26
Retrieving Invalid Rows.....	26
Contacts.....	26
Save contact.....	26
Add to existing group.....	28
Email unsubscribe.....	29
Email resubscribe.....	29
SMS unsubscribe.....	30
SMS resubscribe.....	30
Delete contact.....	30
Delete contact from group.....	31
Preference Subscribe.....	31
Preference Unsubscribe.....	32
Groups.....	32
Create group.....	33
Update group.....	33
Add to existing contact.....	34
Empty group.....	35
Delete group.....	35
Delete contact from group.....	36
MessageDesigns.....	36
Create email message design.....	36
Update email message design.....	38
Delete message design.....	39
Campaigns.....	40
Create email single send campaign.....	40
Update an email single send campaign settings.....	42
Queue a campaign to send.....	44
Cancel a queued campaign.....	45
Delete an existing Campaign.....	45
Admin and Settings.....	46
API Request Logs.....	46
<b>Examples.....</b>	<b>48</b>

**Error Handling ..... 49**

# Introduction

## Purpose

This document provides you with details of the requests that can be made to the REST API to retrieve or commit data to the e-shot™ marketing platform.

It details:

- how requests are authenticated
- what information can be retrieved from the system
- the flexibility you have in querying data from the system
- details of how to interpret the \$metadata (a self-describing document)
- and provides examples of all possible data operations

## Audience

This document is of most benefit to **developers** and **testers** who wish to connect to the REST API.

# Background

Representational State Transfer (REST) is a design architecture that offers a simple way to organise interactions between independent systems, enabling you to interact with e-shot™ from any application that can make an HTTP request.

Our REST API supports OData (Open Data Protocol) for all **read** operations. OData enables the creation of REST-based data services, which allow resources, identified using Uniform Resource Identifiers (URLs) and defined in a data model, to be published and edited by Web clients using simple HTTP messages. OData is a self-describing API. The `/$metadata` endpoint returns an XML document that you can use to determine all entities that can be retrieved from the system, as well as how the entity model can be 'walked' to retrieve child or parent entities within a single request.

The API uses built-in HTTP features, like HTTP authentication and HTTP verbs, which can be understood by off-the-shelf HTTP clients, and supports cross-origin resource sharing to allow you to interact securely with the API from a client-side web application.

If you want a structured, extensible, and simple way to get data out of e-shot™ over HTTP, you want to use our REST API. For all its simplicity the REST API can feel quite complex at first, and we will attempt to break it down into smaller components so that we can easily piece together the full puzzle.

# Getting Started

## Overview

The REST API adheres to the OData standard for **reads** with all requests to write data carried out by making specifically defined POST, PUT, PATCH or DELETE requests.

Using OData, you can control:

- which fields are returned,
- include one or more parent and/or child entities in a single request,
- filter data returned using a range of operators and functions,
- control how many rows are returned,
- control which rows of data within a resultset are returned (page data),
- sort the data

OData is also self-describing enabling users to be able to walk the model structure and work out which entities and attributes they are interested in returning.

## Quick Start

In order to get straight into working with the REST API, navigate to our test bed at:

<https://rest-api.e-shot.net/docs/swagger/index.html>

Then carry out the following activities:

1. Enter the authorization header that you will be using to test the REST API calls

Click on “Authorization”, enter the authentication header value; e.g. “Token F2E2717A-86A6-1661-8DCD-77AE31BD44F7” and click “Authorize”. This will make your header available for the duration of the browser session so that you can quickly and easily test REST API calls.

2. Select a definition

In order to then test an API call, first select from the “Select a definition” dropdown in the top right of your screen the type of operation you wish to perform; e.g. “Data Reads”.

3. Search for the endpoint you wish to test

Enter into the “Filter by Tag” search box the endpoint you wish to test, e.g. “Campaigns”, (the search criteria is case-sensitive). This will filter the available endpoints

4. Click on the endpoint you wish to test

This will expand details on the endpoint, describing the purpose of the endpoint, typically with example request and response structures shown

5. Click on “Try it out”

This will enable you to set any available input parameters and test the request by clicking on “Execute”

6. Click on “Execute”

This will call the REST API endpoint using the credentials you set when you clicked on “Authorise”.

## The 10 most common operations:

### Data Reads

- |                   |                  |
|-------------------|------------------|
| 1. Query Contacts | /Contacts/Export |
| 2. Query Groups   | /Groups          |

### Data Modifications

- |  |   |
|--|---|
| 3. Import Contacts                         | /ContactImports/Upload                          |
| 4. Saving Contacts                         | /V2/Contacts/Save                               |
| 5. Unsubscribe a contact from email        | /Contacts/UnsubscribeEmail                      |
| 6. Adding a contact to a group             | /Groups({key})/Contacts/\$ref                   |
| 7. Removing a contact from a group         | /Groups({key})/Contacts/\$ref?\$id={relatedKey} |
| 8. Subscribe a contact to a preference     | /Contacts({key})/PreferenceSubscribe            |
| 9. Unsubscribe a contact from a preference | /Contacts({key})/PreferenceUnsubscribe          |
| 10. Create a Group                         | /Groups   |

## Data Reads

These are all carried out using the HTTP GET Verb and support the OData protocol.

The URI for OData requests can be broken down into three significant parts:

- Service root URL
- Resource path
- Query options

### Example URL:

The below URL will return the account and subaccount level information.

```
https://rest-api.e-shot.net/Accounts?$expand=Subaccounts
```



The only OData request that is not defined in the \$metadata model is to export Contacts from the system via the /Contacts/Export endpoint. This is because this endpoint has a structure definition that is controlled to some degree by users of the e-shot™ system via the Field Manager.

## Data Modifications

Requests to trigger a system event, or commit changes to the system. Examples include creating a group, deleting a contact or queuing a send.

### Creating new records

HTTP Post requests are used to create new records, or carry out complex business operations such as; importing contacts into the system or queuing a campaign send.

### Updating records

HTTP PATCH / HTTP PUT

Both HTTP Patch and HTTP Put requests are used to update attributes of an entity.

### Deleting records

HTTP Delete requests are used to delete a single entity or collection of a single entity type from the system.

## Authentication

Requests to the REST API are authenticated using Token authentication.

### Token Authentication

A token is passed as part of the “Authorization” header parameter and enables you to be able to make requests against your e-shot account. Please contact our support team who will be able to provide you with a token to use for your requests.

#### Example:

```
Authorization: Token DDFD2EFE-E405-48CC-8A74-AE13BEEB118D
```

NOTE: The “Content-Type” or “Accept” header should be used with all requests. Please see the [Formatting](#) section for more details.

## Testing requests

Please visit the following URL to test your requests.

<https://rest-api.e-shot.net/docs/swagger/index.html>

# Settings

## System Wide Settings

The following system wide defaults have been enforced:

- Accept and return JSON by default\*
- **\$top** has been defined with a **10,000** item limit on responses to requests
- **Max page size** has been defined with a **5,000** item limit
- All date and time field values are in the UTC format
- Expansions are allowed up to 5 levels deep to avoid infinite recursion issues

## Formatting

The REST API supports the following formats:

- JSON
- XML
- CSV\*

The format you wish to supply the request or receive the response can be specified in either of the following:

- Header
- Querystring

The querystring will take precedence over the header if both are supplied.

## Header

### Accept

The “Accept” header must be specified on all GET requests in order to control the accepted output format of the request.

**Allowed values:**

- application/json
- application/xml

**Example:**

```
Accept: application/xml
```

## Content-Type

The “Content-Type” header must be specified on; POST, PUT, PATCH and DELETE requests in order to control the accepted input and output format of the request.

### Allowed values:

- application/json
- application/xml

### Example:

```
Content-Type: application/xml
```

## Querystring

The “\$format” header must be specified in order to control the accepted output or input format of the request.

### Allowed values:

- application/json
- application/xml
- text/csv\*

### Example:

```
https://rest-api.e-shot.net/Accounts?$format=application/xml
```

NOTE: The CSV format is currently only supported on calls to the /Contacts/Export endpoint.

# OData HTTP GET Functionality

All data is read from the REST API using the OData protocol.

Requests made using OData are incredibly flexible allowing one to be able to customise the following for a request:

- The format of the data returned (\$format),
- the fields that are to be returned (\$select, \$expand),
- the subset of data returned (\$filter operators and functions),
- the rows of data within a resultset that are returned (\$top, \$skip),
- the sorting of the returned data (\$orderby)

## Resource Paths

All of the resource paths that are available through the REST API are listed below.

These can also be determined by making a request to /\$metadata or /\$metadata-extended.

\* The /Contacts/Export end point differs slightly from the rest of the OData endpoints due to the data structure for Contacts being entirely customisable by the client via the use of custom fields and as such, it's entity structure is not defined within the \$metadata. Instead the custom field definitions can be determined from the /ContactFieldMappings endpoint.

## Pathing Structure

The URL paths available are defined in the /\$metadata and /\$metadata-extended end points.

Paths can be built to retrieve single entity or multiple entities.

For example, the following query will only display the Account with an ID number of 1:

```
https://rest-api.e-shot.net/Accounts(1)
```

whilst, the below will return all Subaccounts:

```
https://rest-api.e-shot.net/Subaccounts
```

Pathing supports the ability to be able to select an entity or collection of entities relative to a parent or child object.

For example, the following query lists all Subaccounts for the Account with an ID of 1.

```
https://rest-api.e-shot.net/Accounts(1)/Subaccounts
```

whilst the below will return The Subaccount with an ID of 7 associated with the Account with an ID of 1:

```
https://rest-api.e-shot.net/Accounts(1)/Subaccounts(7)
```

Only two levels can be searched in this way and thus the following queries will **not** work:

```
https://rest-api.e-shot.net/Accounts(1)/Subaccounts(7)/Sends
```

```
https://rest-api.e-shot.net/Accounts(1)/Subaccounts(7)/Sends(3)
```

The \$expand query option can provide you enormous flexibility and control of which child or parent entities to return relative to the entity defined by the path. See the [Expansions](#) section for more details.

## Query Options

### Expansions

The \$expand query specifies which child or parent entities are to be included in the resultset (expanded). The following query will display each of the Contacts, along with their expanded Group information:

```
https://rest-api.e-shot.net/Contacts/?$expand=Groups
```

Multiple items can be expanded in a single query if separated by a comma, the following example will display each of the Contacts, with their Group and Source information expanded:

```
https://rest-api.e-shot.net/Contacts/?$expand=Groups,Source
```

Alternatively, all of the items available can be expanded with the following query:

```
https://rest-api.e-shot.net/Contacts/?$expand=*
```

The OData API also supports nested expansions. The below example retrieves the Account, Subaccount and Subaccount statistics information.

```
https://rest-api.e-shot.net/Accounts/?$expand=Subaccounts($expand=Statistics)
```

## Nested Expansions

Nested expansions are supported up to 5 levels deep to avoid infinite recursion.

It is worth noting that **filtering cannot be applied to a nested expansion element and will be ignored**. You can however use \$select, \$orderby, \$top and \$skip within a nested expansion.

## Filter

The \$filter query allows you to restrict the data returned by a query. Filter queries are commonly used with various operators and functions to return the required information, further details of the different types are shown in the following sections.

In the following example, all contacts with a first name equal to “Mark” will be returned:

```
https://rest-api.e-shot.net/Contacts/?$filter=Firstname eq 'Mark'
```

Multiple operators can be used in a single request:

```
https://rest-api.e-shot.net/Contacts/?$filter=(Firstname eq 'Mark' and Lastname eq 'Cavendish') or (Firstname eq 'Miguel' and Lastname eq 'Indurain')
```

Filtering on nested collections is supported, allowing you to filter on a specific row in a nested collection:

```
https://rest-api.e-shot.net/Contacts?$filter=Groups/any(f:f/ID eq 379) and ModifiedDate gt 2017-12-01T16:35:26Z and ModifiedFromSourceID eq 4&$top=10000&$orderby=ModifiedDate desc
```

Below is a comprehensive list of all operators and functions supported by the \$filter query option, with examples.

## Logical Operators

Operator	Description	Query Option Example
eq	Equal	&\$filter=Name eq 'Hawco'
ne	Not equal	&\$filter=Name ne 'Hawco'

gt	Greater than	&\$expand=Statistics&\$filter=Statistics/TotalHardBounces gt 0
ge	Greater than or equal	&\$expand=Statistics&\$filter=Statistics/TotalHardBounces ge 0
lt	Less than	&\$select=ID,TotalSent&\$filter=TotalSent lt 5
le	Less than or equal	&\$select=ID,TotalSent&\$filter=TotalSent le 5
and	Logical and	&\$filter=TotalSent lt 5 and Status eq 'Active'
or	Logical or	&\$filter=TotalSent lt 5 or Status eq 'Active'
not	Logical negation	&\$filter=not startswith(Name,'Message')

## Arithmetic Operators

Operator	Description	Query Option Example
add	Addition	&\$filter=TotalSent add 5 gt 10
sub	Subtraction	&\$filter=TotalSent sub 5 gt 10
mul	Multiplication	&\$filter=TotalSent mul 5 gt 100
div	Division	&\$filter=TotalSent div 5 gt 100
mod	Modulo	&\$filter=TotalSent mod 2 eq 0

## Grouping Operators

Operator	Description	Query Option Example
( )	Precedence grouping	&\$filter=TotalSent le 10 sub (20 add 11)

## Operator Precedence

OData services use the following operator precedence for supported operators when evaluating \$filter and \$orderby expressions. Operators are listed by category in order of precedence from highest to lowest. Operators in the same category have equal precedence.

Category	Operator	Description
Grouping	( )	Precedence grouping
Primary	/	Navigation
	has	Enumeration flags



	xxx ( )	Method call
Unary	-	Negation
	not	Logical negation
	cast ( )	Type casting
Multiplicative	mul	Multiplication
	div	Division
	mod	Modulo
Additive	add	Addition
	sub	Subtraction
Relational	gt	Greater than
	ge	Greater than or equal
	lt	Less than
	le	Less than or equal
	isof	Type testing
Equality	eq	Equal
	ne	Not equal
Conditional AND	and	Logical and
Conditional OR	or	Logical or

## String Functions

Function	Query Option Example
contains	&\$filter=contains(FieldName,'FieldValue')
startswith	&\$filter=startswith(FieldName,'FieldValue')
endswith	&\$filter=endswith(FieldName,'FieldValue')
length	&\$filter=length(FieldName) eq 8
indexof	&\$filter=indexof(FirstName,'mark') gt -1
substring	&\$filter=substring(FirstName,1,3) eq 'ark'
concat	&\$filter=concat(FirstName,LastName) eq 'MarkHawkins'

## Date Functions

Function	Query Option Example
date	&\$filter=date(Fieldname1) ne date(Fieldname2)
year	&\$filter=year(Fieldname) eq 2017
month	&\$filter=month(Fieldname) eq 12
day	&\$filter=day(Fieldname) eq 8

## Time Functions

Function	Query Option Example
time	&\$filter=time>LastLoggedInDate) eq 16:24:57.6470000
hour	&\$filter=hour(Fieldname) eq 1
minute	&\$filter=minute(Fieldname) eq 0
second	&\$filter=second(Fieldname) eq 0
fractionalseconds	&\$filter=fractionalseconds(Fieldname) eq 0.123

## Math Functions

Function	Query Option Example
round (decimal)	&\$filter=round(FieldWithDecimalValue) eq 10
round (double)	&\$filter=round(FieldWithDecimalValue) eq 10d
floor (decimal)	&\$filter=floor(FieldWithDecimalValue) eq 11
floor (double)	&\$filter=floor(FieldWithDecimalValue) eq 11d
ceiling (decimal)	&\$filter=ceiling(FieldWithDecimalValue) eq 12
ceiling (double)	&\$filter=ceiling(FieldWithDecimalValue) eq 12d

## Type Functions

Function	Query Option Example
isof	&\$filter=isof('Model.AutomatedSeriesCampaign')

## Orderby

The `$orderby` query allows you to request resources in either ascending or descending order, by appending **asc** or **desc** to the specified order field. If neither is specified, then the resources will be displayed in ascending order. In the following example the Campaigns are ordered by Type, in descending order:

```
https://rest-api.e-shot.net/Campaigns/?$orderby=Type desc
```

## Select

The `$select` query allows you to control which fields are to be returned for an entity. The following example will return only the email address for each Contact:

```
https://rest-api.e-shot.net/Contacts/?$select=Email
```

Additional fields can be added as required, separated by a comma. Alternatively all fields can be selected by using an asterisk. The following query will display the Email address, Company name and Telephone number for each contact:

```
https://rest-api.e-shot.net/Contacts/?$select=Email,Company,Telephone
```

The query result will display the specified fields in the selected order. The `$select` query option can be specified within nested expansions to restrict the fields returned by child and parent entities included in an expansion.

```
https://rest-api.e-shot.net/SendContacts/?$filter=Campaign/LastSendDate gt 2012-12-03T07:16:23Z&$expand=Contact($select=ID,Email),Campaign($select=Name,SubaccountID,FirstSendDate,LastSendDate;$expand=Statistics($select=TotalClicks,TotalDisplays,TotalUnsubscribes,TotalHardBounces,TotalSoftBounces))&$top=5000&$skip=15000
```

## Top

The `$top` query option parameter is used to restrict the number of items returned from a request.

The following query will display the first 5 Contacts:

```
https://rest-api.e-shot.net/Contacts/?$top=5
```

If `$orderby` is included in the query, the data will be ordered on the server before returning the specified number of items.

In the following example, all of the Contacts will be ordered by the Contact "Lastname" in descending order with the top 5 then returned to the requesting application:

```
https://rest-api.e-shot.net/Contacts/?$top=5&$orderby=Lastname desc
```

Custom page sizes can be implemented through this query, rather than the default results being returned.

## Skip

The \$skip query option is responsible for skipping over the specified number of items from the results to be returned.

The following query will skip the first 10 Contacts and display from the 11<sup>th</sup> item onwards, ordered by ID in ascending order:

```
https://rest-api.e-shot.net/Contacts/?$skip=10&$orderby ID
```

The \$skip query can be used to navigate through pages of data when used in conjunction with the \$top.

## Paging

The \$top query allows a custom number of items to be returned, used in conjunction with the \$skip query option, you can control which 'page' of data is to be returned from the query. It is a good idea to use this feature to control the size of the payload and thus time taken for the request to complete.

If the \$top value is specified to a value greater than the maximum allowed number of items returned from a query, an exception will be raised by the application.

If the \$top value is specified to a value greater than the maximum page size, the resultset will include a URI that can be used to pull the next data page.

## Count

The \$count query allows you to request a count of items returned in the response.

Note the slightly different URL structure:

```
https://rest-api.e-shot.net/Contacts/$count
```

The \$count should always go after the resource path in the URL even though any \$filter queries are processed beforehand, as in the following query:

```
https://rest-api.e-shot.net/Contacts/$count?$filter=Firstname eq 'Mark'
```

If a \$count is used, \$expand and \$select query options will be ignored.

# Custom Requests

The following are a list of REST API reads and writes that are not best described with OData.

## Login

```
GET: /Identity/EshotLogin
```

This request is responsible for returning a link to the calling code that can be used to automatically log into e-shot™. This enables customers to be able to seamlessly transition from your own UI into the e-shot™ website. Each link works by providing a one-time token that expires in 24 hours.

There are two different types of queries, each generating a link that will log the user into e-shot™.

The first query link will just log the user into e-shot™ and navigate the user to either:

- The Subaccount selection page for an account with multiple subaccounts
- The Subaccount Dashboard for an account with a single subaccount

The request below can be used to generate a URL that will log the user into e-shot™. Taking the user to the same destination as would be experienced by manually logging into an account.

```
https://rest-api.e-shot.net/Identity/EshotLogin?ApiKey={token}
```

Example response:

```
https://console.e-shot.net/Login.aspx?Token=a8bb55c1-4aa4-445a-9c2c-77d19702dde5
```

The request below can be used to generate a URL that will log the user into a specific e-shot™ subaccount. Taking the user to the same destination as would be experienced by manually logging into an account and selecting a specific subaccount.

```
https://rest-api.e-shot.net/Identity/EshotLogin?&SubaccountID=7&ApiKey={token}
```

Example response:

```
https://console.e-shot.net/Login.aspx?Token=a8bb55c1-4aa4-445a-9c2c-77d19702dde5&SubAccountID=7
```

# Imports

POST: /ContactImports/Upload

This request allows you to upload contacts in bulk into e-shot™ from a .csv, .xlsx, or .xls file. The first row of the uploaded file must be a header row with columns containing the exact names of one or more of the standard or custom fields available from within your account. If a column is named differently in your file to these fields then the import will be rejected. Upon making a valid request to the API the client will be returned a token which uniquely relates to the import. This token can then be used to track the progress of an import or return contact data relating to any invalid rows not processed during the import.

When importing a CSV file, each row must be separated by a line feed character and a carriage return; \r\n.

Below is an example jQuery \$ajax() request to demonstrate how to build a request. Due to the fact the POST is composed of key/value pairs and a file, the “enctype” of the request must be “multipart/form-data”. This is to indicate that text and binary data is to be passed to the server and to specify the encoding type of the form data. The FormData object in the example instantiates with all the name/value pairs in the specified HTML form including the file upload input.

```
$.ajax({
  type: 'POST',
  enctype: 'multipart/form-data',
  url: 'https://rest-api.e-shot.net/ContactImports/Upload',
  processData: false,
  contentType: false,
  async: false,
  cache: false,
  data : new FormData($('#formId')[0]),
  beforeSend: function (xhr) {
    xhr.setRequestHeader ("Authorization", "Token " + apiKey);
  },
  success: function(response) {
    alert(response.Token);
  },
  error: function (e) {
    alert(e.statusText + ':' + e.responseText);
  }
});
```

## Request Parameters

Name	Description	Type	Required	Default
ImportAction	The action to perform on the imported Contacts. <u>Possible values:</u> <ul style="list-style-type: none"> <li>• Import</li> <li>• Repopulate</li> <li>• Unsubscribe</li> </ul>	Enum	No	Import
SubaccountId	The id of the Subaccount in which to import the Contacts	Integer	Yes	
SourceId	The id of the source which the Contacts will import against	Integer	Yes	
GroupIds	The ids of the Group(s) into which the Contacts will import. These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]	Integer Array	Yes	
FileImport	Data to represent the file of Contacts for upload	Byte Array	Yes	
NotificationEmailAddress	The email address of the recipient to be informed of import successes and failures	String	No	
EnableEmailNotifications	If set to "true" the intended recipient will receive email notifications to inform of import successes and failures	Boolean	No	false
OverwriteExistingDataWithImport	A flag to determine whether file data should overwrite the data for existing Contacts	Boolean	No	false
IgnoreEmailAddresses	An array of email address usernames to be ignored during the import process.  These can be a semi-colon separated list of values or a JSON array. E.g. e.g. Sales@; NoReply@ or [e.g. Sales@, NoReply@]	String Array	No	
ChannelPreferenceMappingIds	ChannelPreferenceMappingIds cannot be supplied if the Preference Centre is not enabled  Represents the Preferences the Contacts have subscribed to  These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]  The ChannelPreferenceMappings can be looked up from the /Channels, /Preferences or the /ChannelPreferenceMappings endpoints	Integer Array	No	



## Behaviour

The following describes the different types of behaviours supported by this endpoint. These are controlled by the setting of the ImportAction parameter value.

### Import

This will import the Contacts into the specified Subaccount and Group(s) for the supplied Source.

If a Contact does not exist in the Subaccount it will be added.

If the Contact does exist in the Subaccount and the OverwriteExistingDataWithImport parameter value is set to "true", the Contacts will be updated.

If the Contact does exist in the Subaccount and the OverwriteExistingDataWithImport parameter value is set to "false", the Contacts will remain unchanged.

### Repopulate

This will import the Contacts in the file into the specified Group(s) and remove from the specified Group(s) any Contacts that do not exist in the supplied file.

If a Contact does not exist in the Subaccount it will be added.

If the Contact does exist in the Subaccount and the OverwriteExistingDataWithImport parameter value is set to "true", the Contacts will be updated.

If the Contact does exist in the Subaccount and the OverwriteExistingDataWithImport parameter value is set to "false", the Contacts will remain unchanged.

### Unsubscribe

This will unsubscribe the supplied email addresses from the system and if supplied, add these contacts to the specified groups.

The import file for this import action only requires the Email field to be supplied in the file.

The OverwriteExistingDataWithImport cannot be "true" for a SuppressionList.

## Import Progress

Using the returned unique token from the [/ContactImports/Upload](#) request, a call can be made to get information about the progress of an import. The possible statuses of an import are: "Waiting", "Success" and "Error".

Example request:

```
https://rest-api.e-shot.net/ContactImports ('0063f286-594f-4458-a4ba-8036fc295b9e')
```

## Retrieving Invalid Rows

Once a contacts file has been imported via a call to [/ContactImports/Upload](#). A request is made to retrieve the status of the import via a call to [/ContactImports](#) using the returned upload token. If the ContactImport entity returned has an "Error" import status (ImportStatusID = 2). Then a call to [/ContactImports/InvalidRows](#) will retrieve the error file.

For ease of use the ContentType of the response is "text/csv", which can be written to file and then opened in a text editor or Microsoft Excel.

Example request:

```
https://rest-api.e-shot.net/ContactImports/InvalidRows?token={token}
```

## Contacts

There are a number of different methods to use to add, update or delete contacts from e-shot™.

### Save contact

```
POST: /V2/Contacts/Save
```

This POST method creates or updates a contact in the given subaccount. When successful, the method returns the contact information including the ID that you may need for further operations.

```
--header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  

```

```

--data '{
    "Email": "restapi2@test.com",
    "MobileNumber": "07411197191",
    "SubaccountID": 3,
    "SourceID": 12,
    "Salutation" : "Mr",
    "Firstname" : "Test",
    "Lastname" : "Contact",
    "Telephone" : "02033208777",
    "URN" : "https://e-shot.net",
    "Company" : "Fofront",
    "GroupIDs": [315],
    "ChannelPreferenceMappingIDs" : [1],
    "CustomFields": [
        {
            "DisplayName": "AutomatedText",
            "FieldValue": "custom value for AutomatedText"
        },
        {
            "DisplayName": "AccountNumber",
            "FieldValue": "custom value for AccountNumber"
        },
        {
            "DisplayName": "AccountManagerName",
            "FieldValue": "custom value for AccountManagerName"
        }
    ]
}'

```

The {token} placeholder should be replaced with a valid authentication token.

## Request Parameters

Name	Description	Type	Required	Lookup value from
Email	A unique email address for the contact	String	Yes	
MobileNumber	A valid mobile number for the contact	String	No	
SubaccountID	The ID of the subaccount in which to add the contacts	Integer	Yes	/Subaccounts
SourceID	The ID of the source which the contacts will be added against	Integer	Yes	/Sources
GroupIDs	The IDs of the group(s) into which the contacts should be added. These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]	Integer Array	Yes	/Groups
Salutation	Salutation for the contact	String	No	
Firstname	First name of the contact	String	No	
Lastname	Last name of the contact	String	No	
Telephone	Contact's telephone number	String	No	

URN	Your unique reference for the contact	String	No	
Company	Contact's company name	String	No	
ChannelPreferenceMappingIds	<p>ChannelPreferenceMappingIds cannot be supplied if the Preference Centre is not enabled</p> <p>Represents the Preferences the contacts have subscribed to</p> <p>These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]</p> <p>The ChannelPreferenceMappings can be looked up from the /Channels, /Preferences or the /ChannelPreferenceMappings endpoints</p>	Integer Array	No	/ChannelPreferenceMappings
CustomFields	<p>The list of custom fields and their values defined against the subaccount</p> <p>CustomFields defined against the subaccount can be looked up</p> <p>Either RowNumber or DisplayName should be supplied against all custom fields</p>	String Array	No	/Contacts?\$top=1&\$expand=CustomFields

## Add to existing group

POST: /Contacts({id})/Groups/\$ref

This request adds a contact to an existing group, the method requires a Contact ID and a Group ID from the same subaccount in the following format:

```
curl --request POST \
  --url 'https://rest-api.e-shot.net/Contacts({id})/Groups/$ref' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
  --data '{
    "@odata.id": "https://rest-api.e-shot.net/Groups({groupId})"
  }'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing contact ID.
- {groupId} should be replaced with a valid existing group ID.

## Email unsubscribe

```
POST: /Contacts({id})/UnsubscribeEmail
```

This method unsubscribes the contact's email address, preventing all outgoing emails to them.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Contacts({id})/UnsubscribeEmail' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing contact ID.

## Email resubscribe

```
POST: /Contacts({id})/ResubscribeEmail
```

This method resubscribes the contact's email address. Please contact the e-shot™ support team if you would like to use this method as a specific permission will need to be enabled.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Contacts({id})/ResubscribeEmail' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing contact ID.

## SMS unsubscribe

```
POST: /Contacts({id})/UnsubscribeSms
```

This method unsubscribes the contact's mobile number, preventing all outgoing SMS to them.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Contacts({id})/UnsubscribeSms' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing contact ID.

## SMS resubscribe

```
POST: /Contacts({id})/ResubscribeSms
```

This method resubscribes the contact's mobile number. Please contact the e-shot™ support team if you would like to use this method as a specific permission will need to be enabled.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Contacts({id})/ResubscribeSms' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing contact ID.

## Delete contact

```
DELETE: /Contacts({id})
```

This method deletes an existing contact from all groups.

```
curl --request DELETE \  
  --url 'https://rest-api.e-shot.net/Contacts({id})' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  

```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing contact ID.

## Delete contact from group

```
DELETE: /Contacts({id})/Groups/$ref?$id={groupId}
```

This method deletes an existing contact from a specific group. This method is logically no different to the method with the same name defined on the Groups entity.

```
curl --request DELETE \  
  --url 'https://rest-api.e-shot.net/Contacts({id})/Groups/$ref?$id={GroupID}' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  

```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing Contact ID. This can be looked up from the /Contacts endpoint.
- {GroupID} this can be looked up from the /Groups endpoint.

## Preference Subscribe

```
POST: /Contacts({ContactID})/PreferenceSubscribe
```

This method subscribes a contact to an existing Preference or Preference and Channel.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Contacts({ContactID})/PreferenceSubscribe' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --data '{
```

```
"PreferenceID": {PreferenceID}
}'
```

- {token} placeholder should be replaced with a valid authentication token.
- {ContactID} should be replaced with a valid Contact ID.
- {PreferenceID} should be replaced by a valid Preference ID.

### Request Parameters

Name	Description	Type	Required	Lookup value from
ContactID	The unique ID of the contact	Integer	Yes	/Contacts
PreferenceID	The unique ID of the Preference	Integer	Yes	
ChannelID	The unique ID of the Channel	Integer	No	

## Preference Unsubscribe

```
POST: /Contacts({id})/PreferenceUnsubscribe
```

This method unsubscribes a contact from an existing Preference or Preference and Channel.

```
curl --request POST \
  --url 'https://rest-api.e-shot.net/Contacts({ContactID})/PreferenceUnsubscribe' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --data '{
    "PreferenceID": {PreferenceID}
  }'
```

- {token} placeholder should be replaced with a valid authentication token.
- {ContactID} should be replaced with a valid Contact ID.
- {PreferenceID} should be replaced by a valid Preference ID.

## Groups

There are a number of different methods to use to add, update or delete groups from e-shot™.



## Create group

POST: /Groups

This request creates a new group in the subaccount.

```
curl --request POST \
  --url 'https://rest-api.e-shot.net/Groups' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
  --data '{
    "SubaccountID": 4,
    "Name": "restApiGroup",
    "Description": "Group created from REST api",
    "IsCheckedByDefault": false,
    "IsShownInUI": true,
    "GroupType": "User"
  }'
```

- {token} placeholder should be replaced with a valid authentication token.

### Request Parameters

Name	Description	Type	Required	Lookup value from
Name	The unique name of the group	String	Yes	
SubaccountID	The ID of the subaccount in which to add the group	Integer	Yes	/Subaccounts
Description	Additional information related to the group	String	No	
IsCheckedByDefault	Controls if the group is ticked by default during sign-up interactions	Boolean	No	
IsShownInUI	Should this group be shown in the "Manage Profile" and "Sign Up Form" pages	Boolean	No	
GroupType	This should always be "User"	String	No	

## Update group

POST: /Groups

This method allows an existing group to be updated.

```
curl --request POST \
  --url 'https://rest-api.e-shot.net/Groups' \
```

```
--header 'Accept: application/json' \
--header 'Authorization: Token {token}' \
--header 'OData-Version: 4.0' \
--data '{
  "ID": {id},
  "SubaccountID": 4,
  "Name": "restApiGroup",
  "Description": "Group created from REST api",
  "IsCheckedByDefault": false,
  "IsShownInUI": true,
  "GroupType": "User"
}'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing group ID.

### Request Parameters

Name	Description	Type	Required	Lookup value from
ID	The unique ID of the group	Integer	Yes	/Groups
Name	The unique name of the group	String	No	
SubaccountID	The ID of the subaccount in which to add the group	Integer	No	/Subaccounts
Description	Additional information related to the group	String	No	
IsCheckedByDefault	Controls if the group is ticked by default during sign-up interactions	Boolean	No	
IsShownInUI	Should this group be shown in the "Manage Profile" and "Sign Up Form" pages	Boolean	No	
GroupType	This should always be "User"	String	No	

## Add to existing contact

```
POST: /Groups({id})/Contacts/$ref
```

This method adds an existing contact to a chosen group.

```
curl --request POST \
--url 'https://rest-api.e-shot.net/Groups({id})/Contacts/$ref' \
--header 'Accept: application/json' \
--header 'Authorization: Token {token}' \
--header 'OData-Version: 4.0' \
--data '{
```

```
}' "@odata.id":"https://rest-api.e-shot.net/Contacts({ContactID})"
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing group ID.
- {ContactID} this can be looked up from /Contacts.

## Empty group

```
POST: /Groups({id})/Empty
```

This method removes all the contacts from the selected group.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Groups({id})/EmptyGroup' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0'
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing group ID.

## Delete group

```
DELETE: /Groups({id})
```

This method deletes the selected group. Please note that only a GroupType of “User” can be removed, for all other group types the EmptyGroup method should be used.

```
curl --request DELETE \  
  --url 'https://rest-api.e-shot.net/Groups({id})' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  
  \
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing group ID.

## Delete contact from group

```
DELETE: /Groups({id})/Contacts/$ref?$id={ContactID}
```

This method removes the contact from a selected group. This method is logically no different to the method with the same name defined on the Contacts entity.

```
curl --request DELETE \  
  --url 'https://rest-api.e-shot.net/Groups({id})/Contacts/$ref?$id={ContactID}' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  
  --data ''
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing group ID.
- {ContactID} this can be looked up from /Contacts.

## MessageDesigns

We have method to add update or delete message design from e-shot™.

### Create email message design

```
POST: /MessageDesigns
```

This POST method can be used to create HTML or Text Messages within the supplied e-shot subaccount. e-shot message design contents could be in the form of HTML and/or Text. The HTML content can be loaded into the system from a string or from a URL that returns HTML content. Similarly Text content can be supplied from a string or from a URL containing text. When only HTML content is supplied and not Text content, e-shot will generate the Text content from the HTML content. The method will fail when the content has invalid links or images however when successful it returns the ID of the newly created message design, that may be needed for further operations.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/MessageDesigns' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  
  --data '{  
    "Name": "My First REST Message",  
    "SubaccountID": 3,  
    "SubjectLine": "simple REST subject for a test message",  
  }'
```

```

    "HtmlContent": "<html> <body> <h1>My First Heading</h1> <p>My first
paragraph.</p> </body> </html>",
    "FromName": "From REST API",
    "ReplyEmail": "reply@yourdomain.com",
    "EmailFromAddressID": 6
  }'

```

The {token} placeholder should be replaced with a valid authentication token.

## Request Parameters

Name	Description	Type	Required	Format/Example
Name	A unique name for the message design	String	Yes	Valid string of minimum 3 to maximum 150 characters.
SubaccountID	The ID of the subaccount in which to add the message design	Integer	Yes	Can be looked up from the endpoint: /Subaccounts
SubjectLine	Subject line for the email message	String	Not required if InferSubjectLineFromHtmlTitle is set to true and the Html contains an HTML title element value	Valid string of minimum 2 to maximum 90 characters.
InferSubjectLineFromHtmlTitle	When true, we will extract an appropriate subject line from your HTML	Boolean	No	
HtmlUrl	A valid URL containing the HTML content to be uploaded	String	Only required if HtmlContent is not supplied	Example: http://www.yourhtmlcontent.co.uk?p1=a&p2=b
HtmlContent	Valid HTML string to be used as email content.		Not required if an HtmlUrl is supplied	
TextUrl	A valid URL containing the Text content to be uploaded	String	No	
TextContent	Encoded campaign text for Text emails	String	No	
HtmlSourceRootUrl	Base URL of HTML images and links	String	No	
TextSourceRootUrl	Base URL for text links	String	No	
FromName	The name from email message is received	String	Yes	Valid string of minimum 3 to maximum 255 characters
ReplyEmail	The email address where replies are expected	String	Yes	Valid string of minimum 3 to maximum 255 characters

EmailFromAddressID	The email address that simulate as the message sender	String	Yes	Can be looked up from the endpoint: /EmailFromAddresses
FromNameContactFieldMappingID	Custom Contact ID to be used as FromName	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'
ReplyEmailContactFieldMappingID	Custom Contact ID to be used as ReplyEmail	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'

## Update email message design

```
PATCH: /MessageDesigns({id})
```

This PATCH method can be used to update an existing message design. The method only allows changing the message design settings. If the message design content needs updating, a new message design is to be created.

```
curl --request PATCH \
  --url 'https://rest-api.e-shot.net/MessageDesigns' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
  --data '{
    "ID":{id},
    "Name": "My First REST Message - Updated",
    "SubjectLine":"simple REST subject for a test message - updated",
    "FromName":"From REST API",
    "ReplyEmail":"reply@yourdomain.com",
    "EmailFromAddressID":6
  }'
```

- {token} placeholder should be replaced with a valid authentication token
- {id} should be replaced with a valid existing message design ID

### Request Parameters

Name	Description	Type	Required	Format/Example
------	-------------	------	----------	----------------

ID	A unique identified for the message design	Integer	No	Can be looked up from api path: /MessageDesigns
Name	A unique name for the message design	String	Yes	Valid string of minimum 3 to maximum 150 characters
SubjectLine	Subject line for the message design	String	Yes	Valid string of minimum 2 to maximum 90 characters
FromName	The name from email message is received	String	Yes	Valid string of minimum 3 to maximum 255 characters
ReplyEmail	The email address where replies are expected	String	Yes	Valid string of minimum 3 to maximum 255 characters
EmailFromAddressID	The email address that simulate as the message sender	String	Yes	Can be looked up from the endpoint: /EmailFromAddresses
FromNameContactFieldMappingID	Custom Contact ID to be used as FromName	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'
ReplyEmailContactFieldMappingID	Custom Contact ID to be used as ReplyEmail	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'

## Delete message design

```
DELETE: /MessageDesigns({id})
```

This method deletes an existing message design.

```
curl --request DELETE \
  --url 'https://rest-api.e-shot.net/MessageDesigns({id})' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid message design ID.

# Campaigns

We have methods to add, update or delete Single Send Email Campaigns and also methods to Queue, Cancel and Delete Campaigns from the e-shot™ REST API.

## Create email single send campaign

```
POST: /Campaigns/EmailSingleSend
```

This POST method creates a campaign from an existing Message Design and associates it with the specified contact groups. The created campaign can only be queued once via the /Queue endpoint to send an email campaign to the contacts that exist in the specified groups.

```
curl --request POST \  
  --url 'https://rest-api.e-shot.net/Campaigns/EmailSingleSend' \  
  --header 'Accept: application/json' \  
  --header 'Authorization: Token {token}' \  
  --header 'OData-Version: 4.0' \  
  --data '{  
    "Name": "My First REST CAMPAIGN",  
    "SubaccountID":3,  
    "MessageDesignID":407,  
    "GroupIDs": [368],  
    "MessageDesignSettingsOverride":{  
      "SubjectLine" : "10 ways to improve your marketing reach",  
      "EmailSenderDetails":{  
        "FromName":"Joe Blogs",  
        "ReplyEmail":"sendreply@yourdomain.com",  
        "EmailFromAddressID":6  
      }  
    },  
    "GoogleAnalytics":{  
      "Source": "REST API Campaign",  
      "Medium": "REST API",  
      "Campaign": "My First REST CAMPAIGN"  
    }  
  }'
```

The {token} placeholder should be replaced with a valid authentication token

### Request Parameters

Name	Description	Type	Required	Format/Example
------	-------------	------	----------	----------------



Name	A unique name for the email single send campaign to be created	String	Yes	Valid string of minimum 3 to maximum 250 characters
SubaccountID	The ID of the subaccount associated with the Campaign	Integer	Yes	Can be looked up from the endpoint: /Subaccounts
MessageDesignID	A valid ID of a MessageDesign to use for the Campaign	String	Yes	Can be looked up from the endpoint: /MessageDesigns
GroupIDs	The IDs of the Group(s) whose contacts are to receive the email campaign. These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]	Integer Array	Yes	Can be looked up from the endpoint: /Groups
ChannelPreferenceMappingIDs	ChannelPreferenceMappingIDs must only be supplied if the Preference Centre is enabled.  Represents the Preferences the Contacts have subscribed to.  These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]	Integer Array	Only required when the Preference Centre is enabled for the subaccount	The ChannelPreferenceMappings can be looked up from: /ChannelPreferenceMappings or /Channels or /Preferences
MessageDesignSettingsOverride	The user can pass this object in order to override default message settings associated with the supplied MessageDesign	Object	No	
SubjectLine	The subject line to be used for the email message	String	No	Valid string of minimum 2 to maximum 90 characters
EmailSenderDetails	The email sender details to use instead of those associated with the supplied MessageDesign	Object	No	
FromName	The From name associated with the from email of the email to be received	String	No	Valid string of minimum 3 to maximum 255 characters
ReplyEmail	The email address where replies are to be sent to	String	No	Valid string of minimum 3 to maximum 255 characters
EmailFromAddressID	The email address to use to simulate as the message sender	String	Yes	Can be looked up from the endpoint: /EmailFromAddresses
FromNameContactFieldMappingID	The contact field whose value for the recipient is to be used as the From name. Contact fields can be looked up from /ContactFieldMappings	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?filter=IsCustomField eq true and DataType eq 'String'

ReplyEmailContactFieldMappingID	The contact field whose value for the recipient is to be used as the Reply email address. Contact fields can be looked up from /ContactFieldMappings	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'
GoogleAnalytics	Optional object takes Google analytics settings, when enabled for the subaccount	Object	No	
Source	Google analytics Source	String	No	Maximum upto 2000 characters
Medium	Google analytics Medium	String	No	Maximum upto 2000 characters
Campaign	Google analytics Campaign	String	No	Maximum upto 2000 characters

## Update an email single send campaign settings

```
PATCH: /Campaigns({id})/EmailSingleSend
```

This PATCH method can be used to update an already created EmailSingleSend campaign. A campaign can only be queued once via the /Queue endpoint to send an email campaign to the contacts that exist in the specified groups.

```
curl --request PATCH \
  --url 'https://rest-api.e-shot.net/Campaigns({id})/EmailSingleSend' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
  --data ' {
    "ID": {id},
    "Name": "My First REST CAMPAIGN - updated",
    "SubaccountID":3,
    "MessageDesignID":456,
    "GroupIDs": [368, 371],
    " MessageDesignSettingsOverride":{
      "SubjectLine" : "10 ways to improve your marketing reach",
      "EmailSenderDetails":{
        "FromName":"Joe Blogs",
        "ReplyEmail":"sendreply@yourdomain.com",
        "EmailFromAddressID":6
      }
    },
    "GoogleAnalytics":{
      "Source": "REST API Campaign",
      "Medium": "REST API",
      "Campaign": "My First REST CAMPAIGN"
    }
  } '
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid existing campaign ID.

## Request Parameters

Name	Description	Type	Required	Format/Example
ID	A unique identified for email single send campaign	Integer	No	Can be looked up from api path: /Campaigns
Name	A unique name for the email single send campaign to be updated	String	Yes	Valid string of minimum 3 to maximum 250 characters
SubaccountID	The ID of the subaccount associated with the Campaign	Integer	Yes	Can be looked up from api path: /Subaccounts
MessageDesignID	A valid ID of a MessageDesign to used in with the campaign	String	Yes	Can be looked up from the endpoint: /MessageDesigns
GroupIDs	The IDs of the Group(s) whose contacts are to receive the email campaign. These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]	Integer Array	Yes	Can be looked up from the endpoint: /Groups
ChannelPreferenceMappingIDs	<p>ChannelPreferenceMappingIDs must only be supplied if the Preference Centre is enabled.</p> <p>Represents the Preferences the Contacts have subscribed to.</p> <p>These can be a comma separated list of values or a JSON array. E.g. 1,2,3 or [1,2,3]</p>	Integer Array	Only required when the Preference Centre is enabled for the subaccount	The ChannelPreferenceM appings can be looked up from: /ChannelPreferenceM appings or /Channels or /Preferences
MessageDesignSettingsOverride	The user can pass this object in order to override default message settings associated with the supplied MessageDesign	Object	No	
SubjectLine	The subject line to be used for the email message	String	No	Valid string of minimum 2 to maximum 90 characters
EmailSenderDetails	The email sender details to use instead of those associated with the supplied MessageDesign	Object	No	

FromName	The From name associated with the from email of the email to be received	String	No	Valid string of minimum 3 to maximum 255 characters
ReplyEmail	The email address where replies are to be sent to	String	No	Valid string of minimum 3 to maximum 255 characters
EmailFromAddressID	The email address to use to simulate as the message sender	String	Yes	Can be looked up from the endpoint: /EmailFromAddresses
FromNameContactFieldMappingID	The contact field whose value for the recipient is to be used as the From name. Contact fields can be looked up from /ContactFieldMappings	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'
ReplyEmailContactFieldMappingID	The From name associated with the from email of the email to be received	Integer	No	Can be looked up from the endpoint: /ContactFieldMappings?\$filter=IsCustomField eq true and DataType eq 'String'
GoogleAnalytics	Optional object takes Google analytics settings, when enabled for the subaccount	Object	No	
Source	Google analytics Source	String	No	Maximum upto 2000 characters
Medium	Google analytics Medium	String	No	Maximum upto 2000 characters
Campaign	Google analytics Campaign	String	No	Maximum upto 2000 characters

## Queue a campaign to send.

```
POST: /Campaigns({id})/Queue
```

This POST method queues an existing campaign to be sent at the specified time.

```
curl --request POST \
  --url 'https://rest-api.e-shot.net/Campaigns({id})/Queue' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
  --data '    '
```

- {token} placeholder should be replaced with a valid authentication token.

- {id} should be replaced with a valid existing campaign ID.

### Request Parameters

Name	Description	Type	Required	Format/Example
ID	A unique identified for the campaign.	Integer	No	Can be looked up from api path: /Campaigns
SendDateTime	A valid future date & time when the email campaign is to be sent	String	Yes	Format: Year-month-dayTHour:Minute:Seconds Example: 2018-10-20T16:21:00

## Cancel a queued campaign

```
POST: /Campaigns({id})/Cancel
```

This method cancels a queued campaign from sending. Campaigns that are in the queue, waiting to be sent will be cancelled.

```
curl --request POST \
  --url 'https://rest-api.e-shot.net/Campaigns({id})/Cancel' \
  --header 'Accept: application/json' \
  --header 'Authorization: Token {token}' \
  --header 'OData-Version: 4.0' \
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid campaign ID.

## Delete an existing Campaign

```
DELETE: /Campaigns({id})
```

This method deletes an existing campaign that is not active, queued or sent.

```
curl --request DELETE \
  --url 'https://rest-api.e-shot.net/Campaigns({id})' \
```

```
--header 'Accept: application/json' \
--header 'Authorization: Token {token}' \
--header 'OData-Version: 4.0' \
```

- {token} placeholder should be replaced with a valid authentication token.
- {id} should be replaced with a valid campaign ID.

## Admin and Settings

There are a number of methods that can be used to administer and set up your account

### API Request Logs

```
Get: /ApiRequestLogs
```

This method lists the requests you have made to the REST API for the provided authentication token

```
curl --request GET \
--url 'https://rest-api.e-shot.net/ \
--header 'Accept: application/json' \
--header 'Authorization: Token {token}' \
```

- {token} placeholder should be replaced with a valid authentication token.

### Request results

Each record in the returned set has the following fields

Name	Description
ID	A unique identified for the logged request.
SubaccountID	When relevant, the subaccountID for the request
UserID	The ID of the User making the request
IsValid	Did the request return successfully
EndPoint	The API method called
IPAddress	The IP Address where the request came from
HttpMethod	The HTTP method of the request; e.g. GET, POST, DELETE
ContentType	The content type of the request; e.g. application/x-www-form-urlencoded

RequestDate	The date and time of the request
-------------	----------------------------------

# Examples

The following are a set of common queries clients have been using.

Description	URL
<p>Exports contacts efficiently from the system, supporting the formats; XML, JSON and CSV. The example is specifically exporting contacts with a custom field value of "Something" for a custom field named "CustomFieldX". Only the fields "Email" and "CustomFieldX" are being included in the results.</p> <p>This endpoint does not support Expansion. You cannot for example, expand "ClickedContacts" from this endpoint.</p>	<p><code>https://rest-api.e-shot.net/Contacts/Export/?\$filter=CustomFieldX eq 'Something'&amp;\$select=Email,CustomFieldX</code></p>
<p>Exports 2000 contacts, including only the specified fields and filtering on the subaccount with an ID value of 3.</p>	<p><code>https://rest-api.e-shot.net/Contacts/Export/?\$top=2000&amp;\$select=Salutation,Firstname,Lastname,Company,ModifiedDate,SubaccountId&amp;\$filter=SubaccountId eq 3</code></p>
<p>Exports 2000 contacts, ignoring the first 2000, including only the specified fields and filtering on the subaccount with an ID value of 3. In other words, this is returning the second page of the resultset where each page is 2000 rows.</p>	<p><code>https://rest-api.e-shot.net/Contacts/Export/?\$top=2000&amp;\$skip=2000\$select=Salutation,Firstname,Lastname,Company,ModifiedDate,SubaccountId&amp;\$filter=SubaccountId eq 3&amp;\$</code></p>
<p>Exports contacts in subaccount with ID equal to 3 where the ModifiedDate is greater than 2018-05-25T00:00:00Z. All fields on defined on the Contact entity will be returned.</p>	<p><code>https://rest-api.e-shot.net/Subaccounts(3)/Contacts/?\$filter=ModifiedDate gt 2018-05-25T00:00:00Z</code></p>



# Error Handling

The following is a list of all the error codes that can be returned for an unsuccessful request.

Status Code	Error Code	Error Name	Details
400	9	Bad Request	The request was not valid. The structure of the body of the message is incorrect, has invalid values or has invalid data types
401	0	Unauthorized	Access was denied, the supplied credentials are invalid
401	1	Unauthorized	Access was denied, the account has expired
401	2	Unauthorized	Access was denied, the account is suspended
403	6	Forbidden	The requested operation is not permitted for the user
404		Not Found	The requested resource was not found, this might be because the URI is malformed, or the resource is deleted
408	7	Request Timeout	The server did not receive a complete request message within the time that it was prepared to wait
422	5	Unprocessable Entity	Missing record when one expected or value mismatch against what was expected, but all supplied values are of correct data types
422	8	Unprocessable Entity	Duplicate record supplied, but all supplied values are of correct data types
422	11	Unprocessable Entity	Container limit exceeded, but all supplied values are of correct data types
429		Too Many Requests	Too many requests have been made in a given amount of time
500		Internal Server Error	An internal error occurred in the server. This might indicate a problem with the request, or might indicate a problem in the server side code
400	12	Bad Request	The request was not valid. The structure of the body of the message is correct but some of the field values are invalid or has invalid data types. Further information may be available in modelState array with each invalid input parameter.
400	10	Missing field value	The request input field value is missing.

400	13	Invalid Format	The request input field value or data type is invalid.
400	14	Object Not Found	The request failed in finding the object for update or delete.
400	15	Invalid Object Id	There is no object with the given identifier or the identifier is incorrect.
400	16	Invalid Subaccount	The subaccount identifier specified in the request is invalid.
400	17	Unexpected Value	The parameter or the value supplied not expected in the request.
422	18	Object In Use	The request to modify object should fail as the object is already in use.
422	31	Missing Html Content	Unable to process request as HTML content is missing.
422	32	Missing Text Content	Unable to process request as text content is missing.
422	33	Missing Subject Line	Unable to process request as email subject is missing.
422	34	Invalid Html Links	Unable to process request as message design has invalid HTML links.
422	35	Invalid Html Images	Unable to process request as message design has invalid HTML images.
422	36	Invalid Html Links And Images	Unable to process request as message design has invalid HTML links and images.
422	37	Invalid Text Links	Unable to process request as message design has invalid text links.
422	38	Invalid Html Structure	Unable to process request as message design has invalid HTML Structure.
400	39	MessageDesign Not Ready To Send	The specified MessageDesign is not ready to send.
422	40	Campaign Does Not Exist	Unable to process request as Campaign no longer exist.
422	41	Campaign Not Ready To Send	Unable to process request as Campaign is not ready to send.
422	42	Campaign Cannot Be Deleted	Unable to process Campaign deletion request. The campaign may be active or has already been used.
422	43	Campaign Has Been Deleted	Unable to process request as Campaign has already been deleted.

422	44	Campaign Is Already Cancelled	Unable to process request as Campaign is already cancelled.
422	45	Campaign Cancelling Is Inprogress	Unable to process request as Campaign cancellation is inprogress.
422	46	Campaign Has Already Been Completed	Unable to process request as Campaign has already been completed.
422	47	No Contacts Found	Unable to find any valid active contacts to send campaign.

If you require any additional assistance contact our support team on 020 3320 8750, email [support@forfront.com](mailto:support@forfront.com) or talk to us on [live chat](#).